~~Physics~~

COO-1469-0182
COO-2118-0007

QUARTERLY TECHNICAL PROGRESS REPORT

January, February, March 1971

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

QUARTERLY TECHNICAL PROGRESS REPORT

January, February, March 1971

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

TABLE OF CONTENTS

Page

Page

1. CIRCUIT RESEARCH

(Supported in part by the Office of Naval Research under Contract N000 14-67-A-0305-0007, W. J. Poppelbaum, Principal Investigator.)

Summary

Bernard Tse has taken over the work in Bundle Processing and describes in his report the combination "repeater/restorer". SABUMA, the earlier bundle project, is being finished off by Trevor Mudge. Yiu Wo's APE report deals with conversion between different number representations. Colormatrix is finally glowing: Hadjistavros has replaced the pellet resistors with microtransistors as heating elements, and he has also modified the method of thermally biasing and insulating the elements. Finally, Panigrahi comments on the camera and monitor specifications for Pentecost.

M. Faiman, editor

1.1 Bundle Processing (Project No. 21)

1.1.1 Bundle Signal Repeater and Restorer

Since the last quarter, it has been decided that, owing to their structural similarities, the bundle signal repeater and the bundle signal restorer should be incorporated into one system. The design of a system of 6 repeater/restorer stages with a bundle size of 63 wires is planned. The block diagram of such a system is shown in Figure 1.

1.1.2 Analog to Bundle Converter of Repeater

The analog input of the bundle signal repeater is to be converted to bundle representation. From this analog signal 63 wires with "+" and "-" signals have to be generated. A design for this A/B converter is shown in Figure 2.

1.1.3 Repeater/Restorer Stages

The mapping scheme used for the system is shown below:

	Repeater	Restorer
"+"	signal level "1"	signal level "1" in bundle 1 of input
"0"	broken wire, no signal	wire with level "0" in bundles 1 and 2 of input
"-"	signal level "0"	signal level "1" in bundle 2 of input

The purpose of the bundle signal repeater/restorer stages is to detect the wires with no signal information, namely the "0" wires, and assign to these wires "+" or "-" signals randomly. A design for this purpose is shown in Figure 3.

Bernard Tse

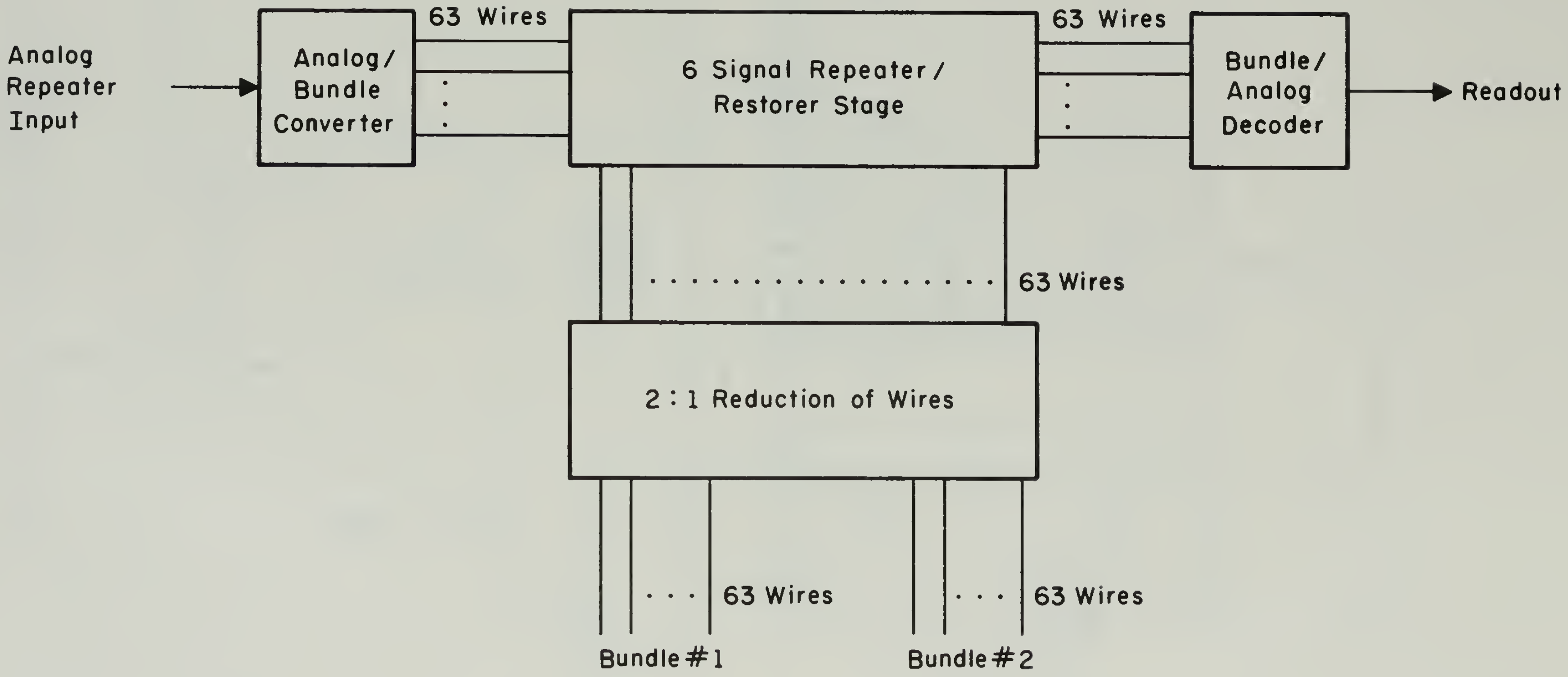


Figure 1. Block Diagram of Bundle Signal Repeater/Restorer

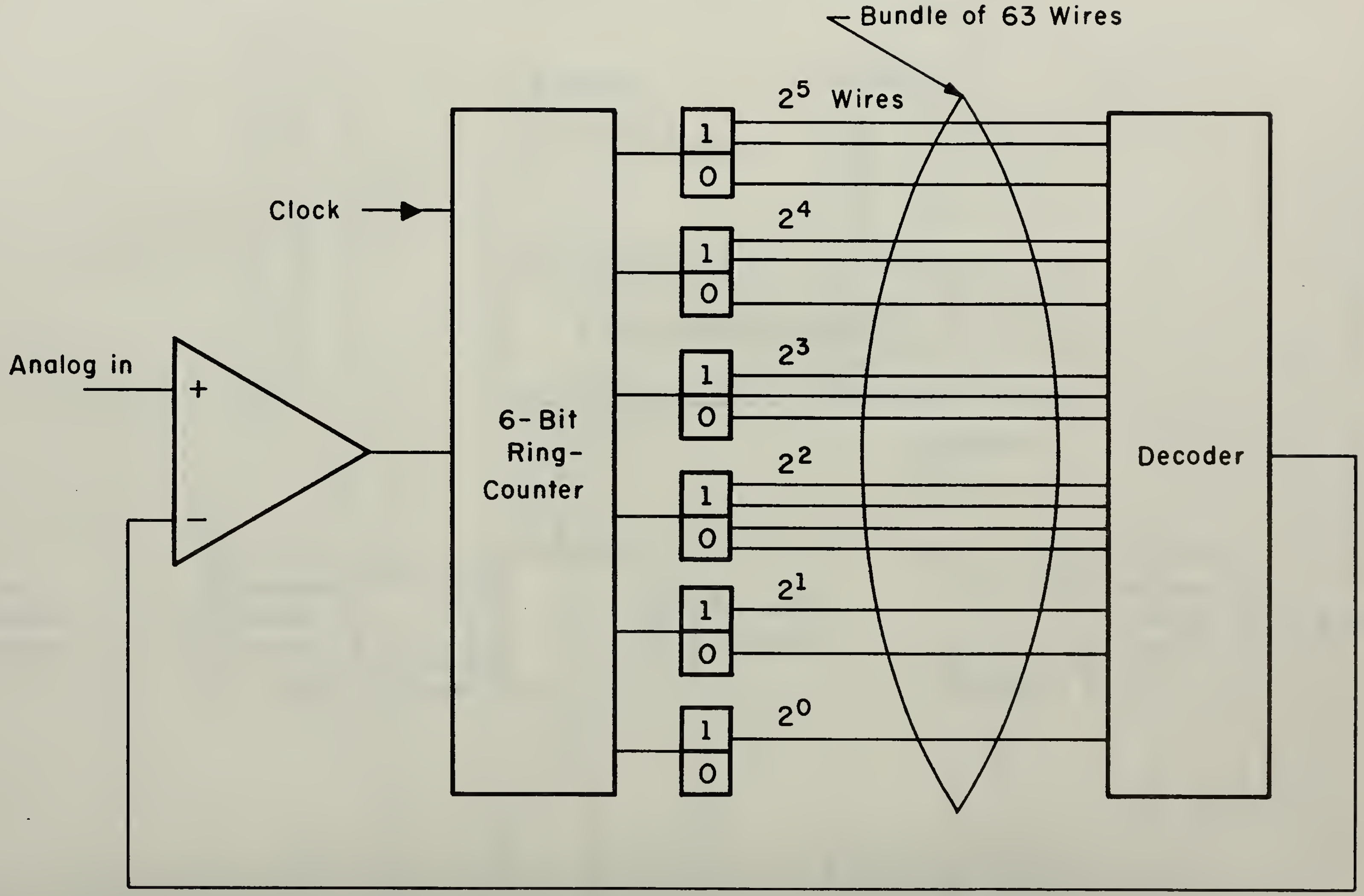


Figure 2. Analog to Bundle (A/B) Converter

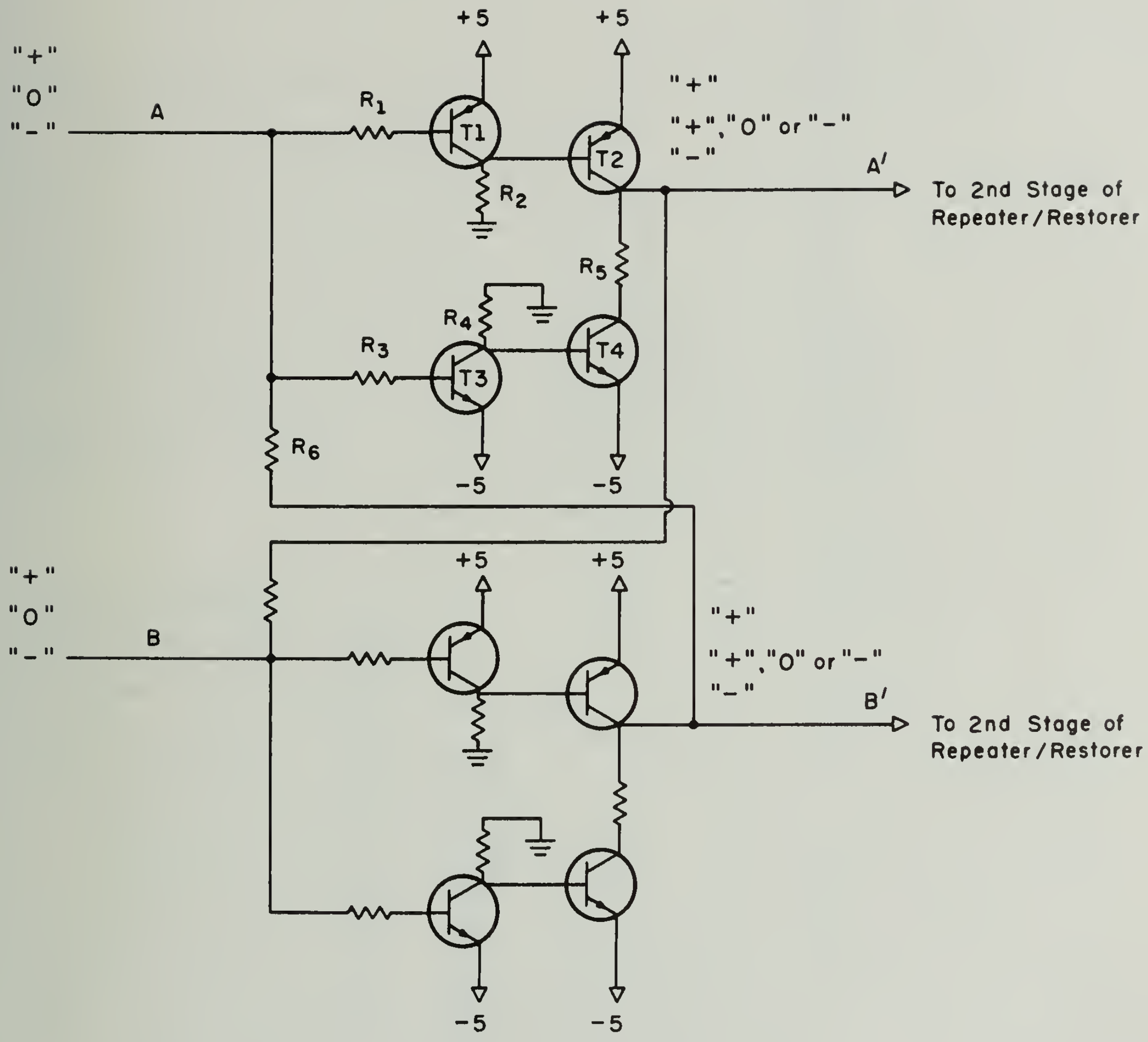


Figure 3. Repeater/Restorer Stage for Two Wires A and B

1.1.4 SABUMA

All printed circuit cards for this project are complete except for the error detectors. The display has been wired but lacks a panel as yet. The only major construction remaining is to build the bundles. The system can then be put under test.

Trevor Mudge

1.2 APE (Project No. 25)

1.2.1 Output Decoding Circuits

In the APE machine, values of the input variables are normalized in the range of -1 to 1. Furthermore, all variables are mapped into machine number representation having a range from 0 to 1 in 10-bit binary accuracy. After they are processed by APEs, the results are sent out by the APEs in machine number representation. For easy readout of the computation results, it becomes necessary to have the output value in machine representation converted back to signed decimal representation. This is done in the control unit of the APE machine by the 1997/1023 scaler together with the machine converter. These units will now be described.

1.2.2 The 1997/1023 Scaler

The output signals from the APEs are in pulse width modulation. The pulse width ranges from 0 to a maximum of 1023 system clock periods. The scaler is to map a number in this range into a number ranging from 0 to 1997, linearly, to the limit of the truncated least significant digit. The idea here is to convert the 10-bit machine number into a 3-digit decimal number plus sign. Figure 1 shows the block diagram of the scaler. What it does is simply to delete one pulse for every forty input pulses. A fast clock having a frequency twice as high as the system clock is used in the input of the scaler to compare with the pulse-width modulated signal from the output of the APEs. With such an arrangement the number of output pulses is that of the input sealed by a factor of 1997/1023.

1.2.3 Machine Number to Signed Decimal Number Converter

Recall that the mapping between these two representations is given by

$$x_i = 1 - 2x_{im} \qquad \begin{array}{c} -1 < x_i < 1 \\ 0 < x_{im} < 1 \end{array}$$

where x_i denotes variable value and x_{im} denotes machine number value.

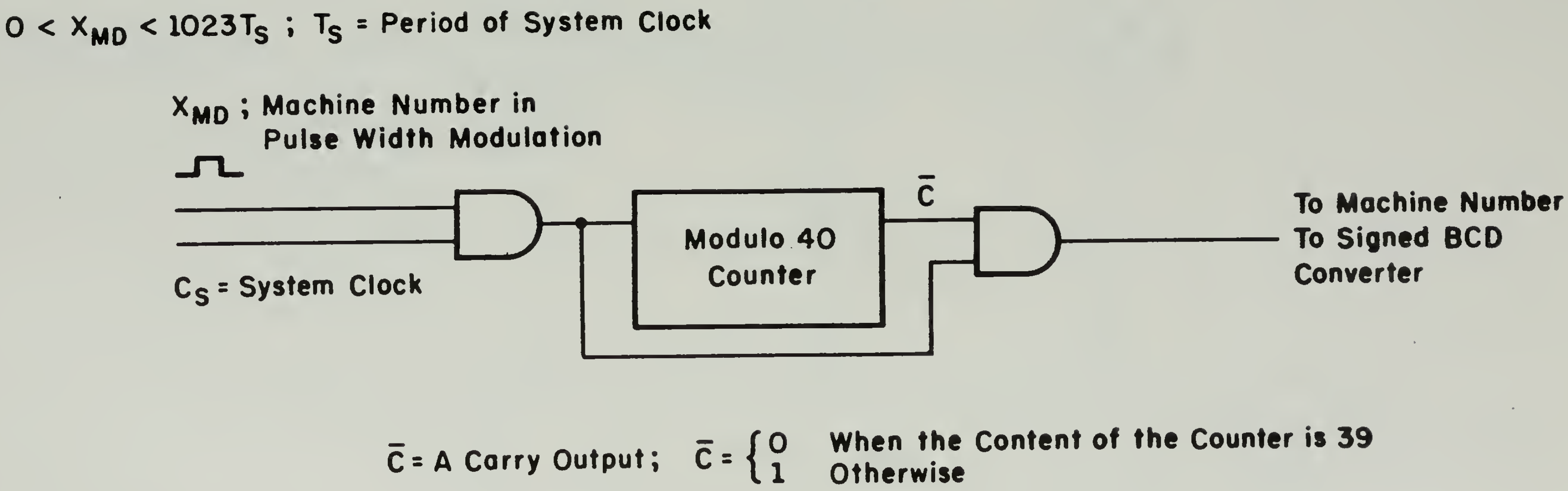

$0 < X_{MD} < 1023T_S$; T_S = Period of System Clock
X_{MD} ; Machine Number in Pulse Width Modulation
C_S = System Clock
Modulo 40 Counter
$\bar{C}$
To Machine Number To Signed BCD Converter
$\bar{C}$ = A Carry Output; $\bar{C} = \begin{cases} 0 & \text{When the Content of the Counter is 39} \\ 1 & \text{Otherwise} \end{cases}$

Figure 1. The 1997/1023 Scaler

The function of the converter is to implement the above relation. This can be carried out by a circuit shown in Figure 2. The decade up-down counters are preset to 999, and the counter is set to count down to begin with. The input is fed by the 1997/1023 scaler. When the contents of the counter reach 000, the counters are switched to count up. If the counting process ends during the count down period, the number is negative, otherwise positive. The magnitude of the signed decimal number is given by the contents of the counters for both positive and negative cases. The full range of the input is 1997 pulses. This corresponds to an output range from -999 to +998. If the decimal point is considered located to the far left, the range of this number is suitable for displaying the variable value between -1 and +1 in 10-bit accuracy.

Yiu Wo

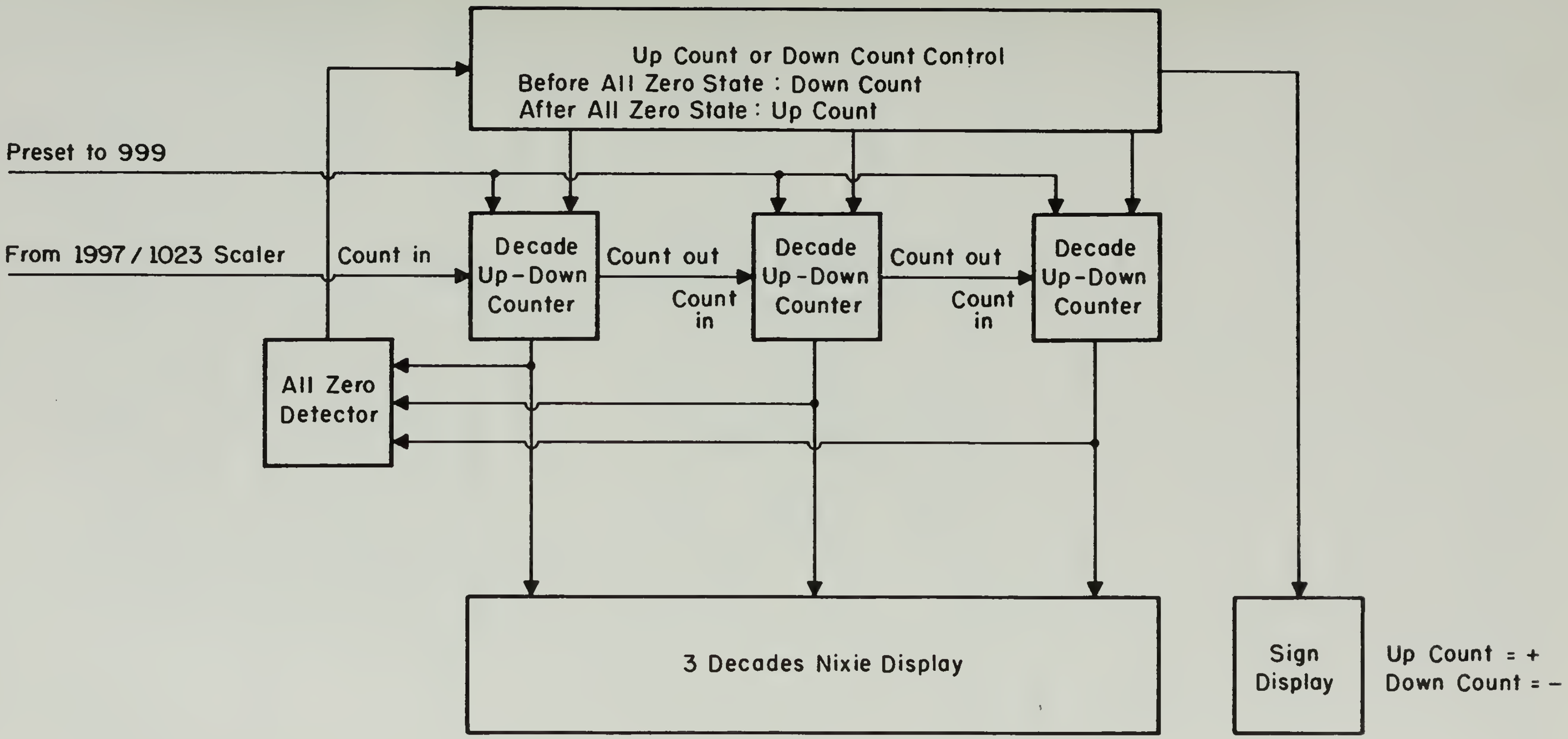


Figure 2. The Machine Number to Signed Decimal Converter

1.3 Colormatrix (Project No. 27)

1.3.1 Achieving Thermal Stability

A 5 × 7 matrix using microtransistors has now been constructed. X-Y addressing is used with the X line corresponding to the emitters, and the Y lines corresponding to the bases, as shown in Figure 1. The corresponding horizontal (X) and vertical (Y) drivers are shown in Figures 2 and 3. Due to the proximity of the transistors in the array, the thermal interaction between neighboring elements causes the active (and thus warmer) elements to heat the non-active elements. Moreover, this average heat, which is added over and above that supplied by the thermal bias, is dependent on the particular character being displayed. Clearly, this additional heat shifts the average temperature above the threshold temperature and makes the displayed characters ambiguous or, at best, destroys the contrast of the display. In the worst case, this shift, ΔT, is approximately 0.5 - 0.6°C. Thus, some sort of thermal isolation is necessary for the elements and, in addition, a sink must be provided for this excess heat. It has been determined that heat loss occurs primarily by two means: by conduction through the metallic leads of the transistors (about 10-20%), and by convection to the air (90-80%). Little can be done about the former, but an attempt was made to reduce the latter by introducing a screen consisting of a thermal insulator laminated to a thermal conductor. The thermally conducting layer was maintained at a much lower temperature than the bias temperature to act as a sink. The thermally insulating layer surrounded the transistors, so as to isolate them and, at the same time, not sink too much heat from them. Using this scheme an improvement of about 50% was noted, that is, ΔT was reduced to 0.2-0.3°C.

Thus, a new approach has been taken. In order to solve this problem, it will be necessary to operate the thermal bias at 1°C below threshold rather than at threshold. Thus, the average increase of 0.2-0.3°C will be insufficient to bring the non-active elements into the thermally active region. This will, of course, be detrimental to the response time. To overcome this we may thermally bias at 1°C below threshold and then apply a pulse to <u>all</u>

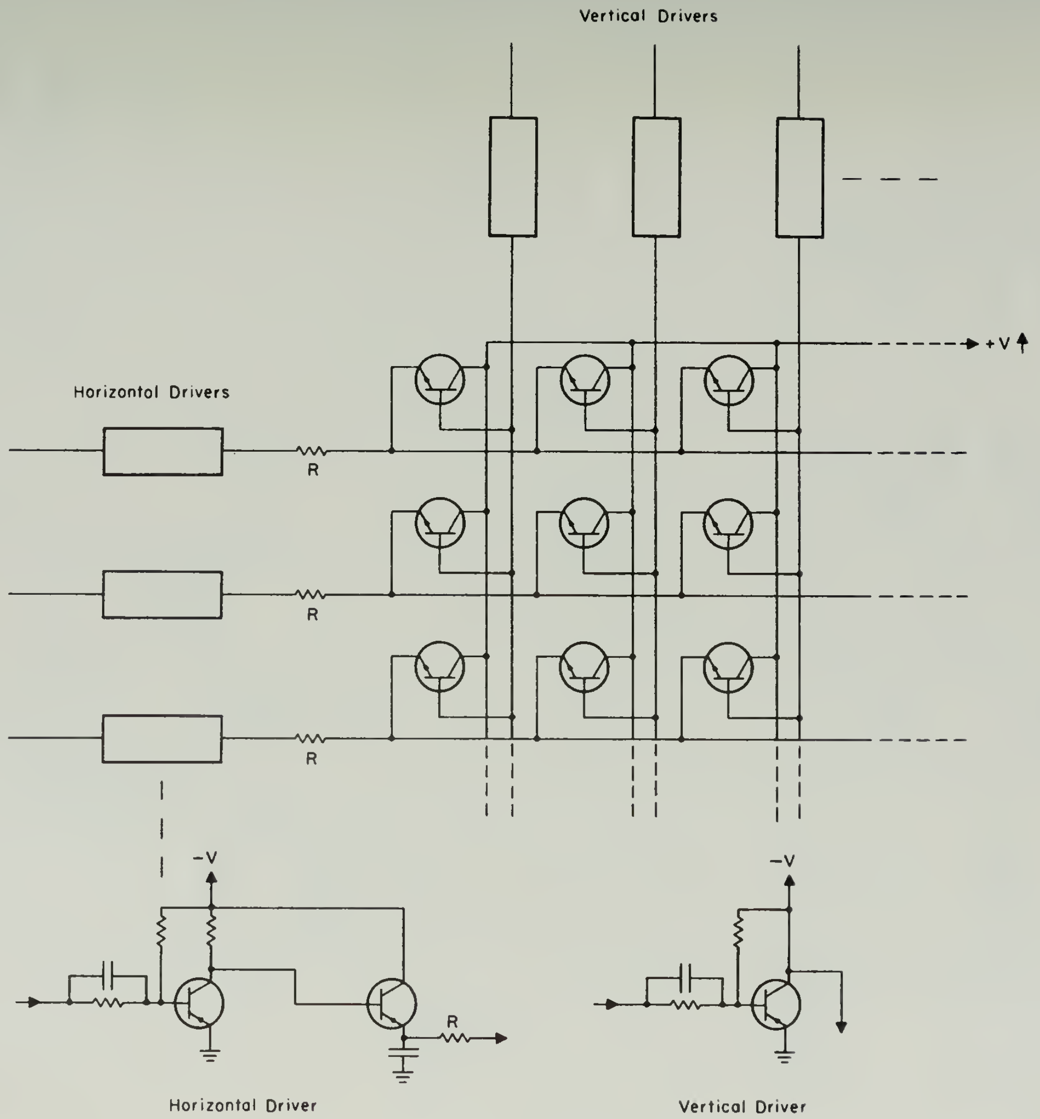


Figure 1. Microtransistor Array

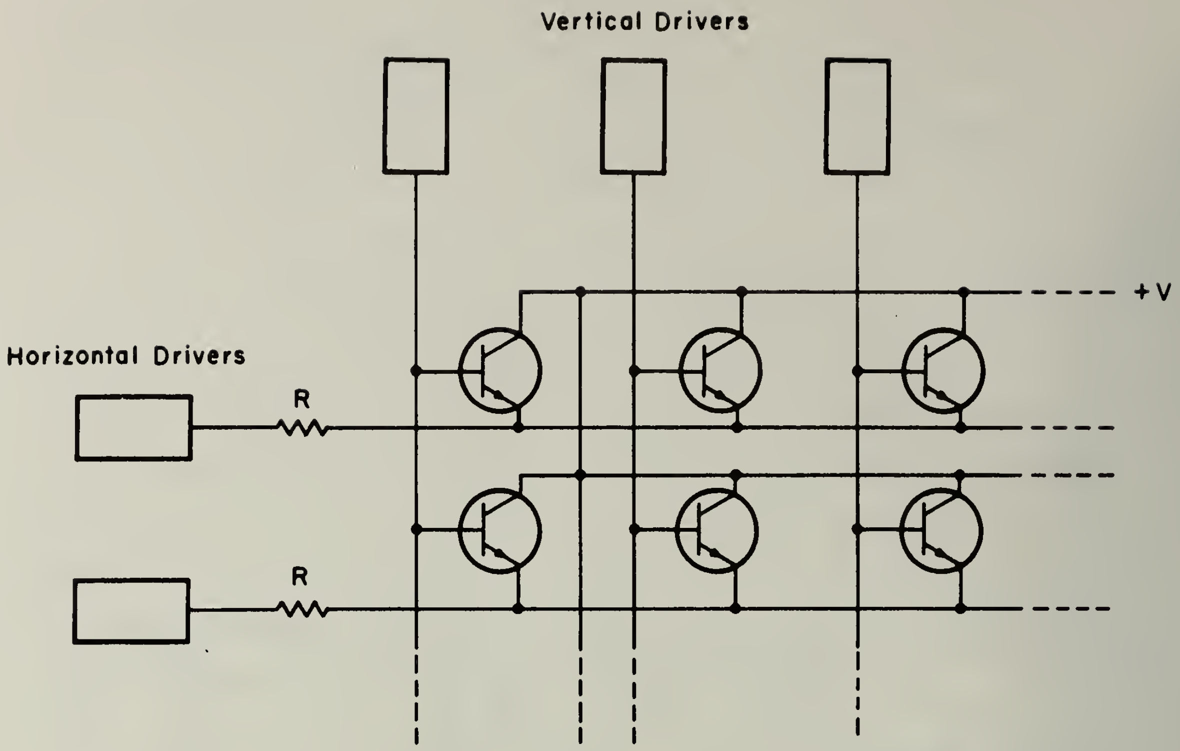


Figure 2. Horizontal Driver

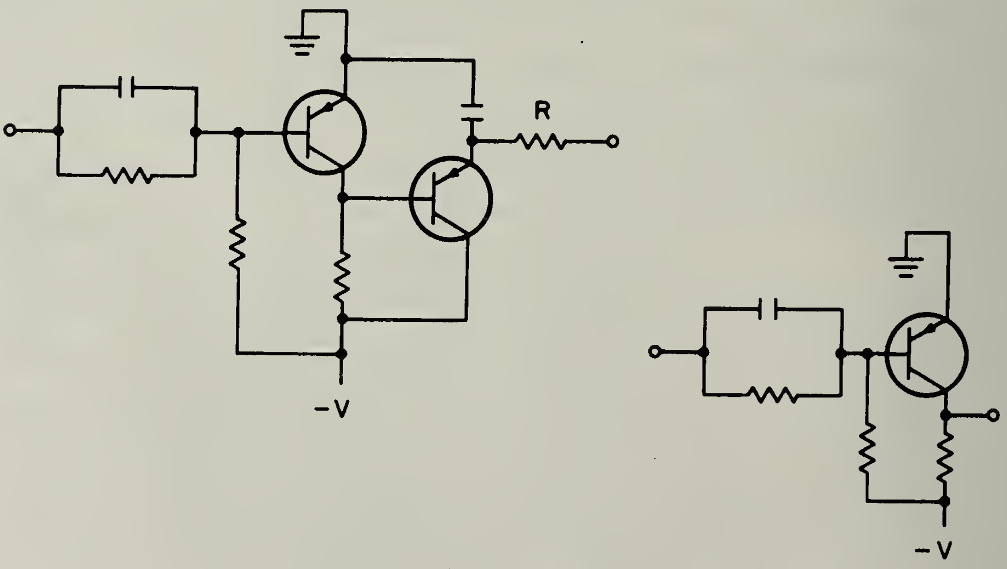


Figure 3. Vertical Driver

elements which is sufficient to bring them up to threshold. When a character is displayed, the active elements receive an additional pulse, and the non-active ones receive no pulse at all. Thus, the active elements are driven up in temperature, and the inactive ones are allowed to "fall" down in temperature. In effect this is a "negative" heat pulse.

One could, of course, operate directly from 34°C by supplying a pulse to the active elements sufficient to bring them up the full 2°C while supplying no pulse to the inactive elements. This has disadvantages, however, in that it is more difficult to adjust the threshold temperature, and, as mentioned before, the response time is slower. Thus, the first method is preferable although more complex.

This method was implemented by dividing the output pulses from the SN7445 decoder (Figure 4), which formerly had a duration 2T, into two pulses of duration T. Elements then receive none, one or two pulses, corresponding to the non-active, standby and active states.

In order to decrease the turn-on time, an additional group of horizontal drivers in Figure 4 (Group B) have been added in parallel with the normal drivers. These supply an additive power pulse during transition periods from one character to another. The duration of the period for which these additional drivers are active is adjustable. The results using this new scheme have been very good.

Stavros Hadjistavros

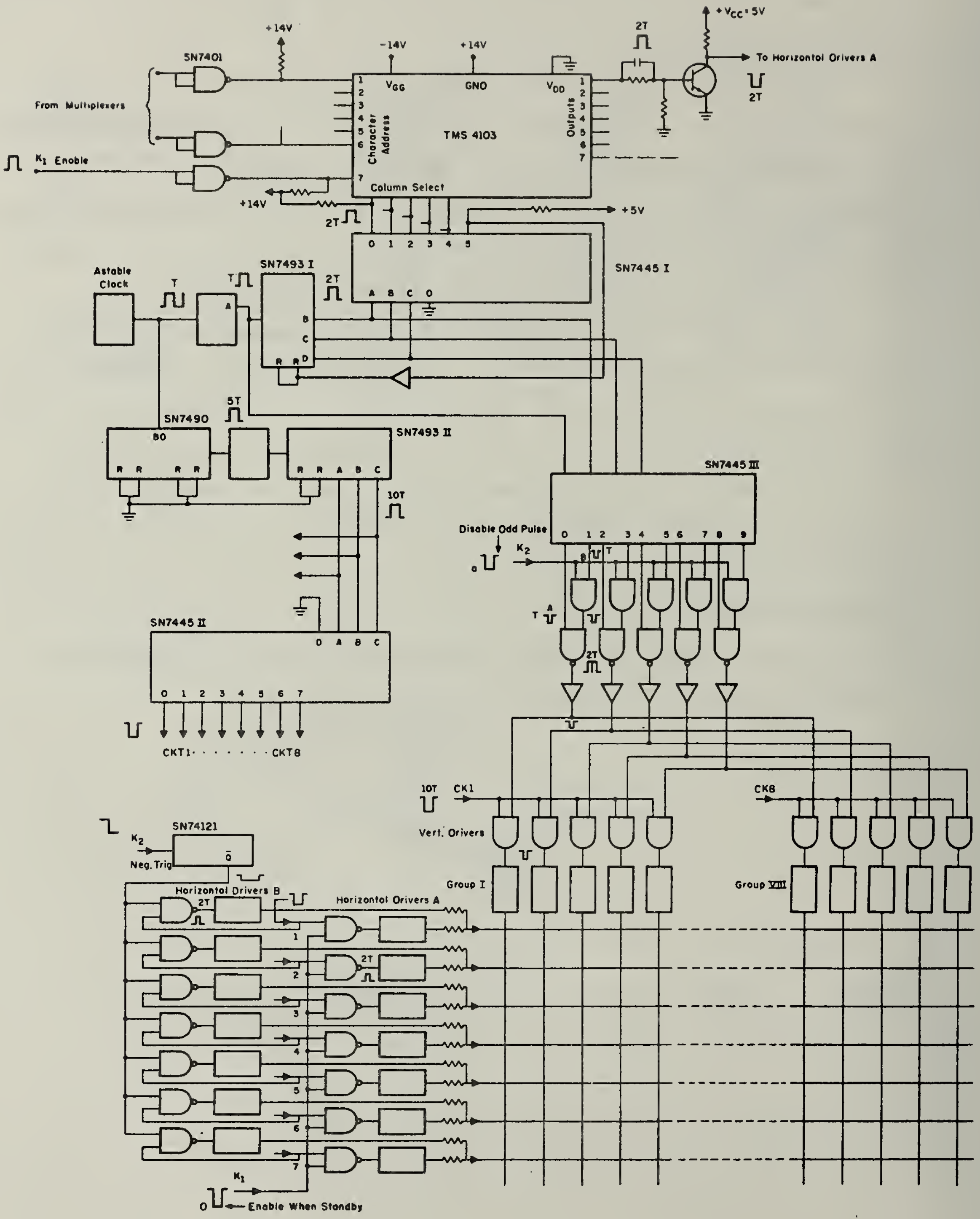


Figure 4. Revised Colormatrix Block Diagram

1.4 PENTECOST (Project No. 31)

1.4.1 Camera

The camera system of PENTECOST has been decided upon. For field-sequential color presentation, it is essential for the camera pick-up tube to have a high speed response, so that there is no carry over from one color field to the other. From cost and performance considerations, it was found that a lead-oxide vidicon (Plumbicon) could satisfy these requirements. Ordinarily, plumbicons do not have good spectral sensitivity in the red region; so we would buy a plumbicon designed to be used in the red channel of color television systems. Red-extended plumbicons were rejected because of cost considerations. The plumbicon tube will be employed in a black and white vidicon/ plumbicon camera.

1.4.2 Monitor

The black and white CCTV monitor and the Penetron tube have been received. The various modifications of the video monitor so as to accommodate the Penetron tube are being considered now.

G. Panigrahi

2. HARDWARE SYSTEMS RESEARCH

(Supported in part by the Atomic Energy Commission under Contract US AEC AT(11-1) 1469, W. J. Poppelbaum, Principal Investigator.)

Summary

M. N. Cooper has concluded that a laser will be required for the LASCOT project, the efficiencies of other types of light source being too low. Progress in the design of the Curie temperature OLFT system is summarized by Doug Sand. Ed Carr's report deals with his redesign of the local oscillator in the ORBIT receiver to achieve synchronization with the transmitter. This project is now almost complete. Don Hanson has details on changes to the Tricolor Cartograph. The BLAST report of Larry Wallman describes problems of amplifying the screen signal. Art Simons has a description of how Eidolyzer's World Model works. The Mudge-Kodimer duo are now building some of the hardware for Semantrix, described last time, and have reached the stage of being able to locate block positions on the table. Dick Blandford's LINDA report deals mostly with his newly designed sweep circuits. Progress with Stereomatrix is written mostly by Dick Cheng, who gives detailed drawings of the Coefficient Generator and Cursor. Dick Partridge describes circuits and polyhedron generation for PAGAN. Finally, Sik Yuen is considering the use of LEDs in the Scantrix project.

M. Faiman, editor

2.1 LASCOT (Project No. 09)

2.1.1 Light Source

The requirement on the light source of LASCOT was for an output beam having maximum divergence of 0.1° and a minimum diameter not exceeding 2.5 mm. The source is required to put out a power of approximately 0.8 watt in blue (440 nm), 0.4 watt in green (510 nm) and 0.3 watt in red (630 nm). Preliminary calculations indicated that for an isotropic source of size s, producing an output beam diameter d with a divergence angle of Φ_3 radians

$$\frac{\text{total luminous flux in output beam}}{\text{total luminous flux emitted by source}} = \frac{1}{16}\left(\frac{d}{s}\cdot\Phi_3\right)^2$$

substituting the following appropriate values: $d = 3$ mm, $s = 4$ mm and $\Phi_3 = 0.1^\circ$, this ratio is about 10^{-7}.

Thus the idea of using an isotropic (or nearly isotropic) source, such as a projection lamp or arc lamp and a collimator, was dropped. The use of a laser is the only logical choice.

Investigations are under way in selecting the most suitable laser.

M. N. Cooper

2.2 OLFT (Project No. 12)

2.2.1 Design of Cooled Assembly

Most of the design details for the vacuum chamber have been specified, and the materials required are being obtained. The modified electron gun is being ordered and the deflection-focus yoke has arrived. Remaining are the crystal assembly (with negotiations for its construction near completion) and the optical components. Design of the electronic circuits has begun with the temperature control circuits now being tested.

Discussions with other research groups working with cooled KDP devices indicate that near the Curie temperature crystal cracking sometimes occurs. To examine this problem and also to test the cooling design, an "intermediate" system is being built, which is essentially the new cooling assembly alone, contained in the old vacuum chamber and using most of the old electronics. Further comments will be included in a subsequent report.

Doug Sand

2.3 ORBIT (Project No. 15)

2.3.1 Modified Synchronization

Until now, the ORBIT receiver has been synchronized with the transmitter by matching a local oscillator of 9.45MHz with the master oscillator. This has proved to be quite difficult. The reduced bandwidth of the system, compared with that of standard TV, made it necessary to count 600 periods of the local oscillator and compare the end of count with the arrival time of the next incoming video line. Any difference was fed back to correct the local oscillator frequency. The circuitry to implement this proved to require critical adjustment, with the result that the synchronization was not always true.

However, the exact value of 9.45MHz is only important for specifying the exact picture width -- an unnecessary parameter, since this width can be adjusted over a wide range on all receivers. The only real requirements on the local oscillator are that (i) it be no less than 9.45MHz to be able to provide 600 time slots in each horizontal line; (ii) its frequency be stable; and (iii) it start in the same phase on arrival of each incoming line of video. This task can be readily accomplished by a very stable, triggerable multivibrator.

The circuit that has been developed is shown in Figure 1, and is actually a retriggerable monostable. The CLEAR AND TRIGGER signal is derived from the synch waveform generator (see last report) and the four inverters provide the delay necessary for retriggering.

Ed Carr

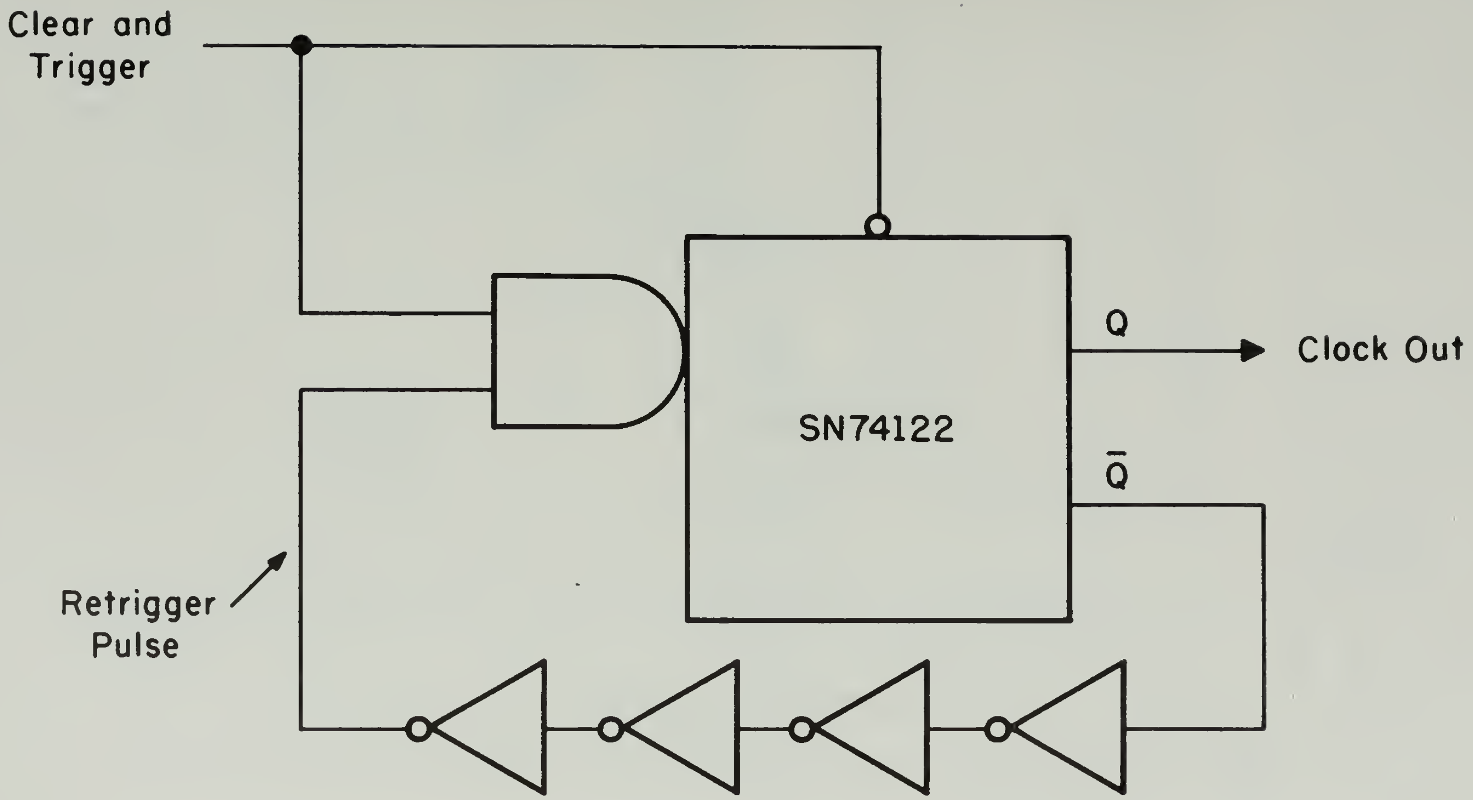


Figure 1. ORBIT Receiver Clock

2.4 Tricolor Cartograph (Project No. 16)

2.4.1 Control Panel Fabrication

Part of the last quarter, was spent in the shop fabricating a suitable control panel out of sheet steel. The switches and slide potentiometers came and the switches are in the process of being assembled and mounted on the control panel. Control wiring was designed to run between the switch panel and the card rack.

2.4.2 New Logic

New logic was designed to make the new switches compatible with the previous logic. A gating circuit was designed and assembled on a PC card. This circuit is the minimal Moore designed circuit for gating one frame of video onto the disk. It is shown in Figure 1.

2.4.3 The Delay Line

The delay line from Corning arrived and was tested in the mode in which it will be used in the project. It seemed to work very well. As yet, no design has been finalized. It will be used in the vertical video to logic converter.

2.4.4 Horizontal Video to Logic Converter

Finally, a bandpass differentiator was designed and assembled on a PC card. This card is used in input of outlines from the blackboard. It is shown in Figure 2.

Don Hanson

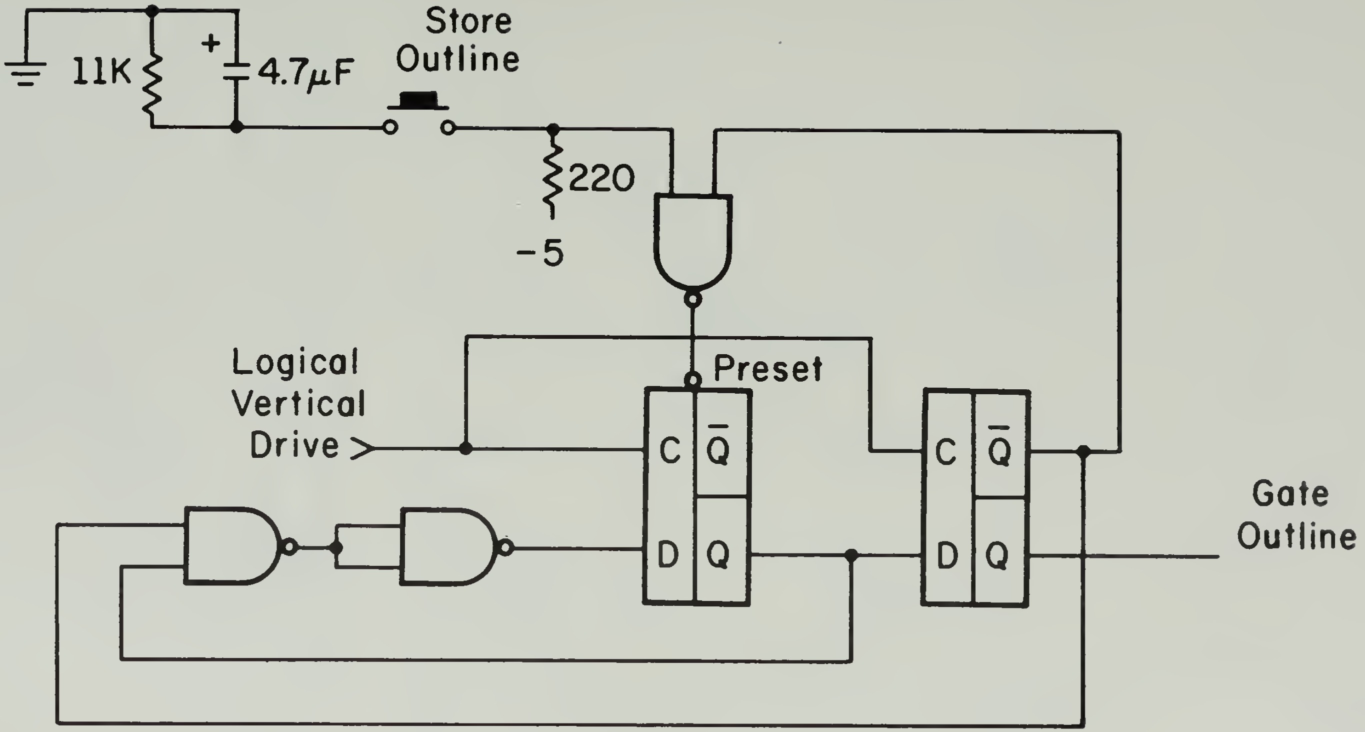


Figure 1. One Frame Gate

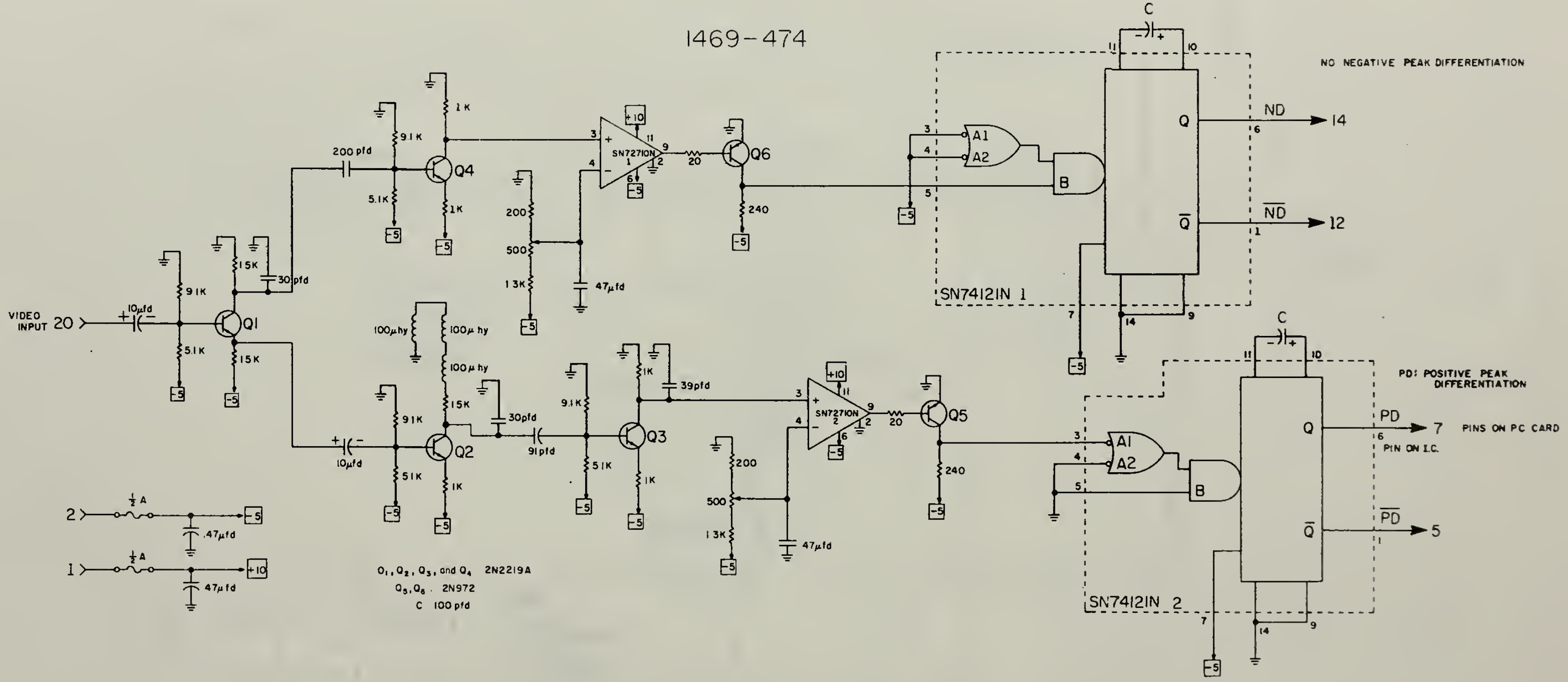


Figure 2. TCC Bandpass Differentiator

2.5 Transformatrix (Project No. 17)

2.5.1 Summary

The second phase of Transformatrix is now complete. Details are to be found in L. Ryan's Ph.D. thesis, "System and Circuit Design of the Transformatrix Coefficient Processor and Output Data Channel", to be published as D. C. L. Report No. 435.

M. Faiman (ed.)

2.6 BLAST (Project No. 19)

2.6.1 Screen Signal

In order to eliminate some of the noise associated with transmitting the very small screen signal down from 18kV, an amplifier has been built which floats at the anode potential. In this way, the signal from the screen is amplified before it is transmitted by the isolation transformer to ground. The common mode noise associated with the high voltage power supply is not amplified as much as the screen signal. The signal, which is obtained using the circuit in Figure 1, is approximately 5mV in amplitude. The electron beam has a constant current for this figure. The screen signal is not uniform in amplitude across a single horizontal line. This is thought to be due to variations in thickness or density of the phosphor coating on the screen. The variation, then, is because the beam does not penetrate the phosphor to the conducting stripes, but rather is a capacitive effect with the phosphor acting as dielectric.

In order to eliminate another source of noise, namely, secondary electrons falling back on the screen, the screen is biased about 10 volts negative with respect to the second or accelerating anode. With this biasing only about 10% of the emitted secondaries fall on the screen. The remaining 90% are collected by the second anode. [1]This biasing is accomplished by simply connecting second anode lead to the positive supply for the amplifier, as seen in Figure 1.

The amplifier, μA733, outputs are connected to emitter followers whose emitters have windings of a transformer in series. In this way any voltage change at the output of the μA733 is immediately transformed

[1] Fundamentals of Television Engineering, Glenn M. Glasford, 1955, McGraw-Hill

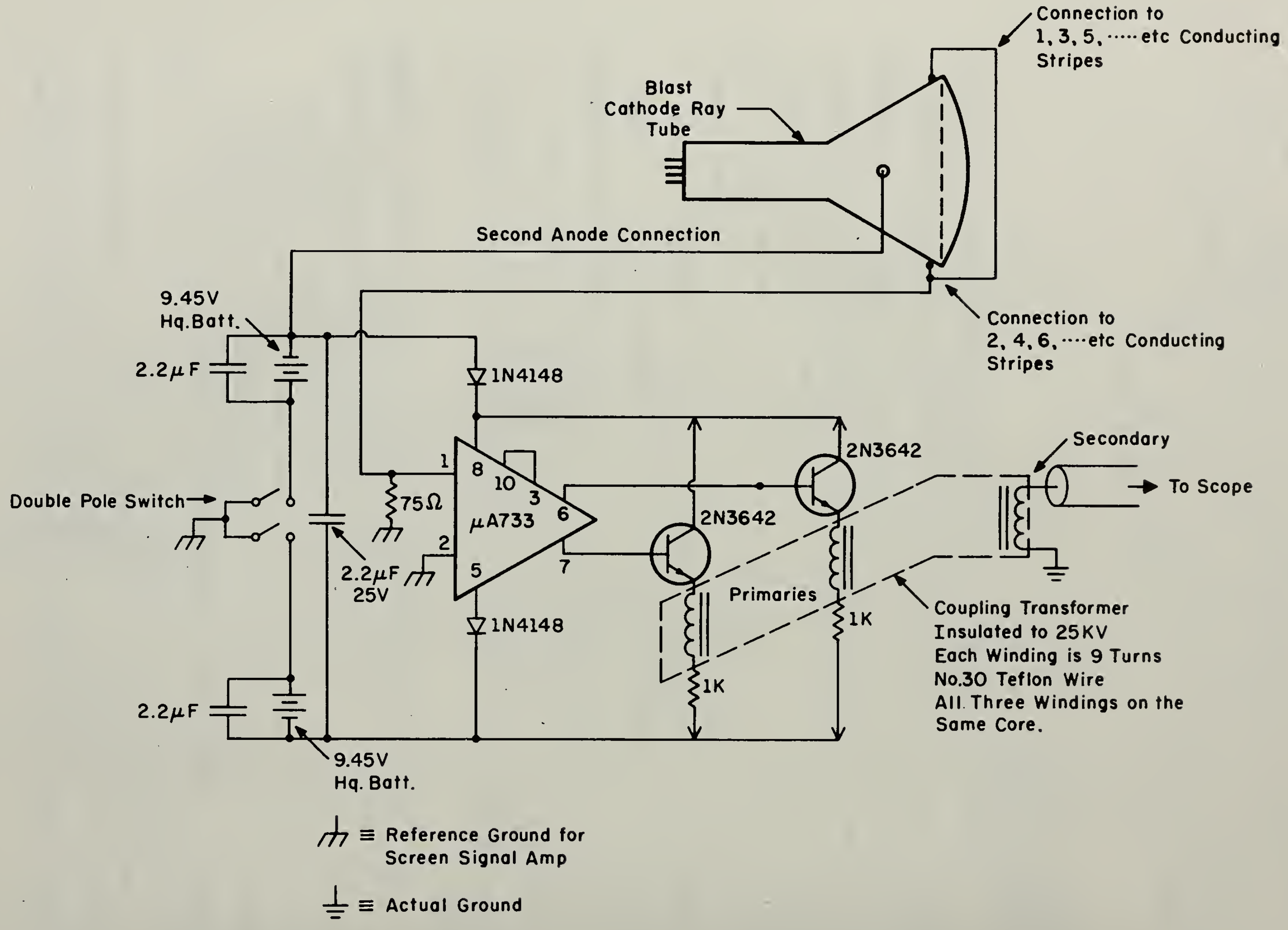


Figure 1. Screen Signal Amplifier

into a current change in the winding and is thereby detected in the secondary of the transformer. The input resistor at pin 1 of the μA733 was determined experimentally.

The immediate goal is to amplify the signal more and to use this signal to synchronize the chopper.

Larry Wallman

2.7 Eidolyzer (Project No. 23)

2.7.1 Progress to Date

As suggested in the previous report, a test jig has been constructed to check out all of Eidolyzer with the exception of the input subsection. Using the test box to simulate the Scan Array, Color Detector, and Row Analyzer (see Figure 1), the remaining subsystems of Eidolyzer have been checked and work perfectly. Furthermore, the prototype card containing the simulated sections has been tested and approval has been given to build the remaining fifteen of these cards. To date, four more have also been built and checked out. The entire system should be constructed and functional during this next quarter.

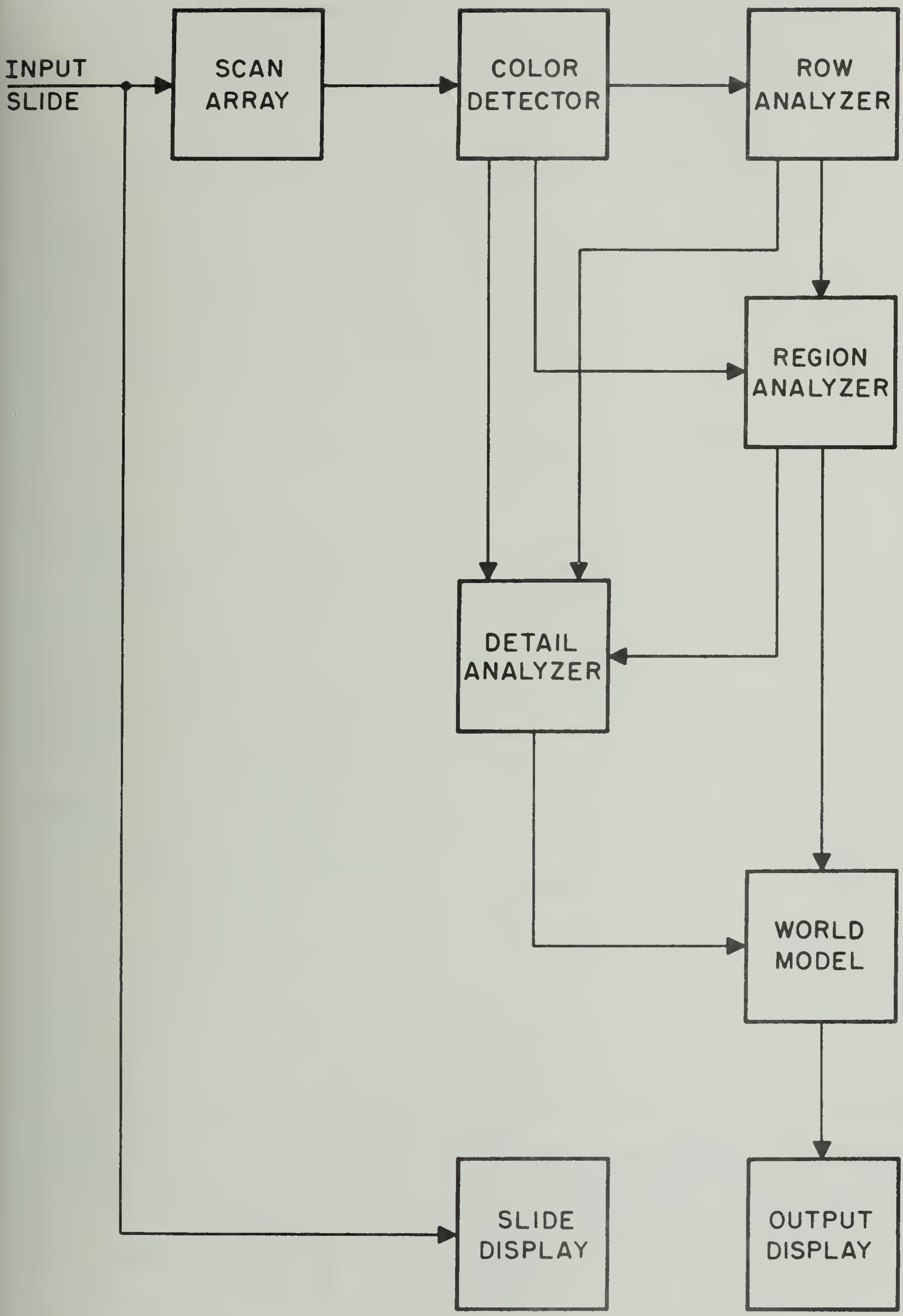


Figure 1. Eidolyzer Block Diagram

2.7.2 Hardware Design

Having demonstrated that the system, with the exception of the input subsection, works as planned, it is appropriate at this time to describe in detail the most important part of the system -- the World Model.

The World Model is constructed with eight standard size printed circuit boards. (See Figure 2) In addition to timing signals, there are sixteen inputs to the World Model per region. There are four possible colors for a region, five possible colors for details, and seven significant combinations of temperature and humidity. The outputs consist of the name for a region from the following seven: SKY, WATER, ROCK, DIRT, SNOW, SAND, GRASS, or zero, one or two names for details from the following twelve:

WATER, SUN, BARN, UFO, GRASS, ISLE, ROCK, DIRT, CLOUD, ICE, SAND, SNOW.

The timing signals provide the necessary sequencing. By considering only certain information at certain times, it is possible to make a decision based solely on that information and modify it later due to newly-arrived information. This process can be repeated as many times as the decision-maker thinks it necessary. The more iterations, the more information considered, and the more accurate the answer. Depending on what accuracy is desired, the decision-maker is free to accept an answer at any point in the process. It is this very technique which is the realization of the layered, hierarchial type structure and which distinguishes the World Model from a simple read-only-memory.

There are four timing signals used in the World Model. The first pre-sets a bank of flip-flops corresponding to the presupposed knowledge of landscapes associated intrinsically with the World Model. This means that, with no other information available, a brown region is hypothesized to be DIRT, a white region to be SNOW, and so on. The second signal gates in the region color information so that the first interpretation can be made. The third timing signal gates in the detail color information. The details are now chosen by a logic array which allows the detail information to flow to a reasonable conclusion through paths biased favorably or negatively by the original region

WORLD MODEL FOR ONE REGION

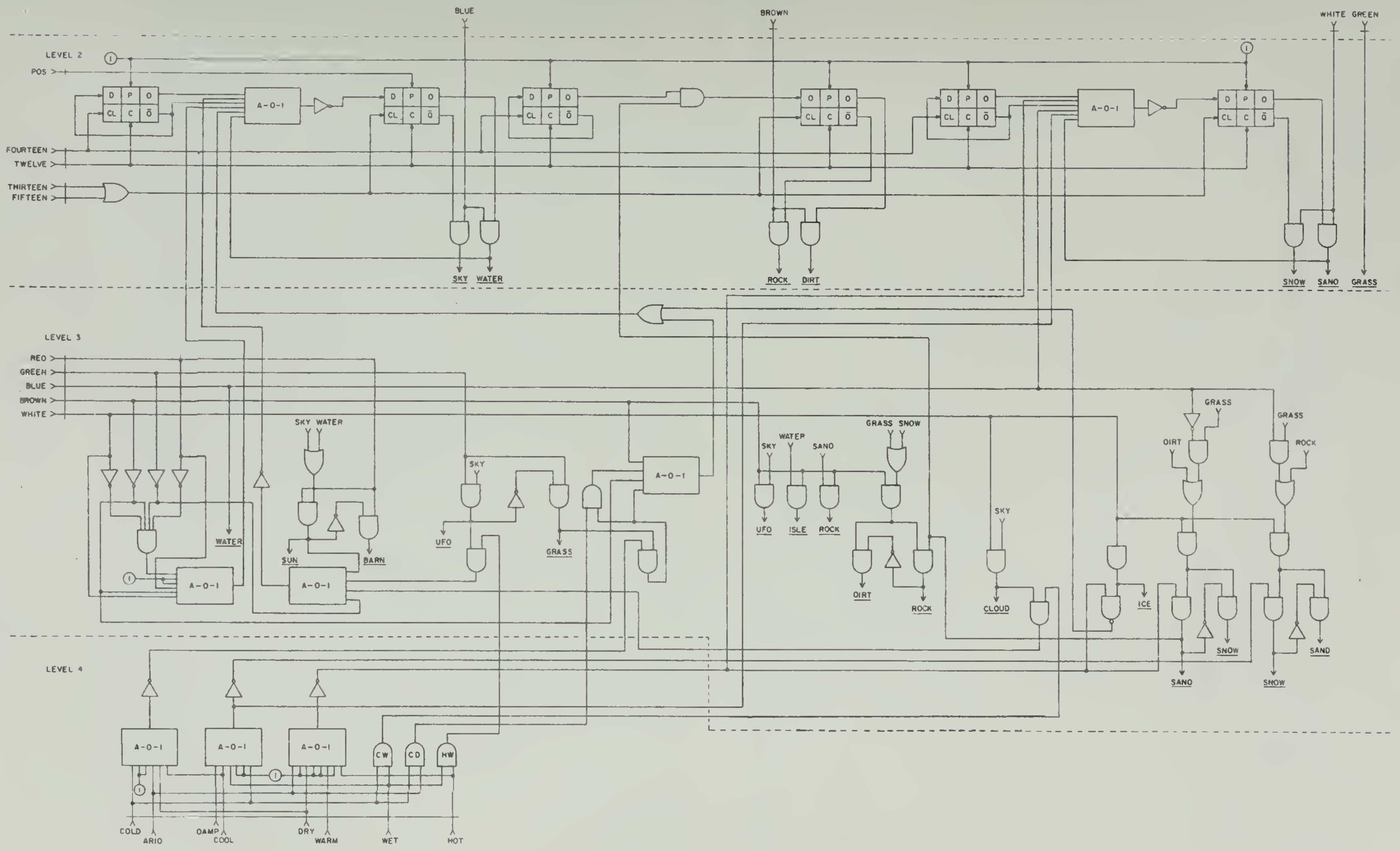


Figure 2. World Model for One Region

hypothesis. For example, a white detail would tend towards SNOW, SAND, ICE or CLOUD but in a region already hypothesized to be SKY only the latter makes much sense. Furthermore, a CLOUD detail tends to increase the "confidence level" of a SKY region and would, in turn, provide some positive feedback to the first level decision. On the other hand, if a SAND detail were found in a region already hypothesized to be DIRT the confidence level would be decreased and a measure of negative feedback would be supplied to the first level decision. Finally, the fourth timing signal is used to add the temperature and humidity information to the decision-making process. This information is supplied by the human observer by means of switches located on the front panel. The four temperature switches read:

HOT, WARM, COOL, COLD

and the four humidity switches read:

ARID, DRY, DAMP, WET.

These switches serve a dual purpose. Their more important function is to provide another level of context for the interpretation, thereby exercising the world model further. A significant combination of temperature and humidity can influence the hypothesis of a region or detail directly, and these in turn can influence the other. Furthermore, the choice of a detail can influence the choice of another detail so that what we have is a fourth level of information capable of influencing third and second level decisions, with both these decisions then capable of influencing other third and second level decisions. The result of all this is a fluidic, nesting-type arrangement where the answer which finally filters through represents the most probable answer based on the presented information. The other function of the temperature and humidity switches is to reflect somewhat the state of mind of the observer. Having just come in out of the bitter cold, a person is perhaps more likely to see a white region as SNOW than a person who has been inside all day. By providing the observer with the opportunity to select a temperature and humidity condition, an environment closer to that of the observer is likely to result.

Art Simons

2.8 Semantrix (Project No. 24)

2.8.1 Summary

The position location scheme described in the last report has been designed. Presently, circuit boards for the sense amplifiers are under construction. A handwired system detecting a 6 × 6 coordinate matrix can be demonstrated. Construction of the major mechanical portions of Semantrix has begun.

Denny Kodimer
Trevor Mudge

2.9 LINDA (Project No. 28)

2.9.1 Summary

The goal of this project is to build a machine which is capable of recognizing a number of simple line drawings. A given line drawing will be displayed on a cathode ray tube via a flying spot scanner. The drawing will then be expanded so that its edges pass through a ring of photocells around the CRT. The information from the photocells will be clocked into registers and decoded to produce an identifying output.

2.9.2 Project Status

Past work has been focused on building a pattern generator and determining a suitable expansion technique. Two pattern generators were built and expansion of each pattern on the CRT was performed by lowering the accelerating voltage across the tube. Results were promising although intensity and focusing problems were encountered during expansion of the patterns.

In January 1971, a flying spot scanner was obtained and work during the past quarter has been directed at building sweep circuits, a one-stage video amplifier, and overcoming the intensity problem introduced by the scanning technique. Figure 1 shows the Synch. Separator Circuit. The vertical synch signal is obtained by integrating the composite signal, while the horizontal signal is obtained by differentiating the composite signal. Figures 2 and 3 show the Vertical and Horizontal Sweep Circuits, respectively. The multivibrators provide for vertical and horizontal sweep even though the synch signal is lost. (Note

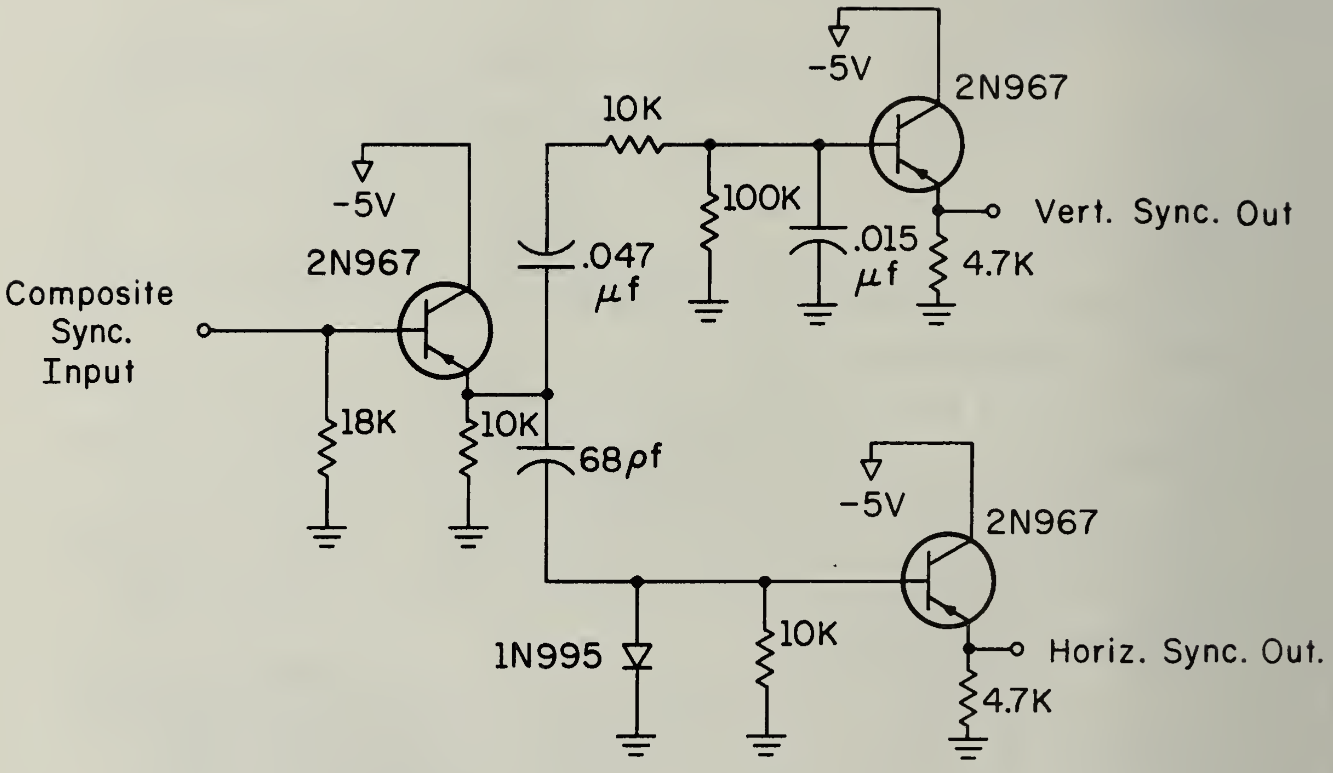


Figure 1. Synch. Separator Circuit

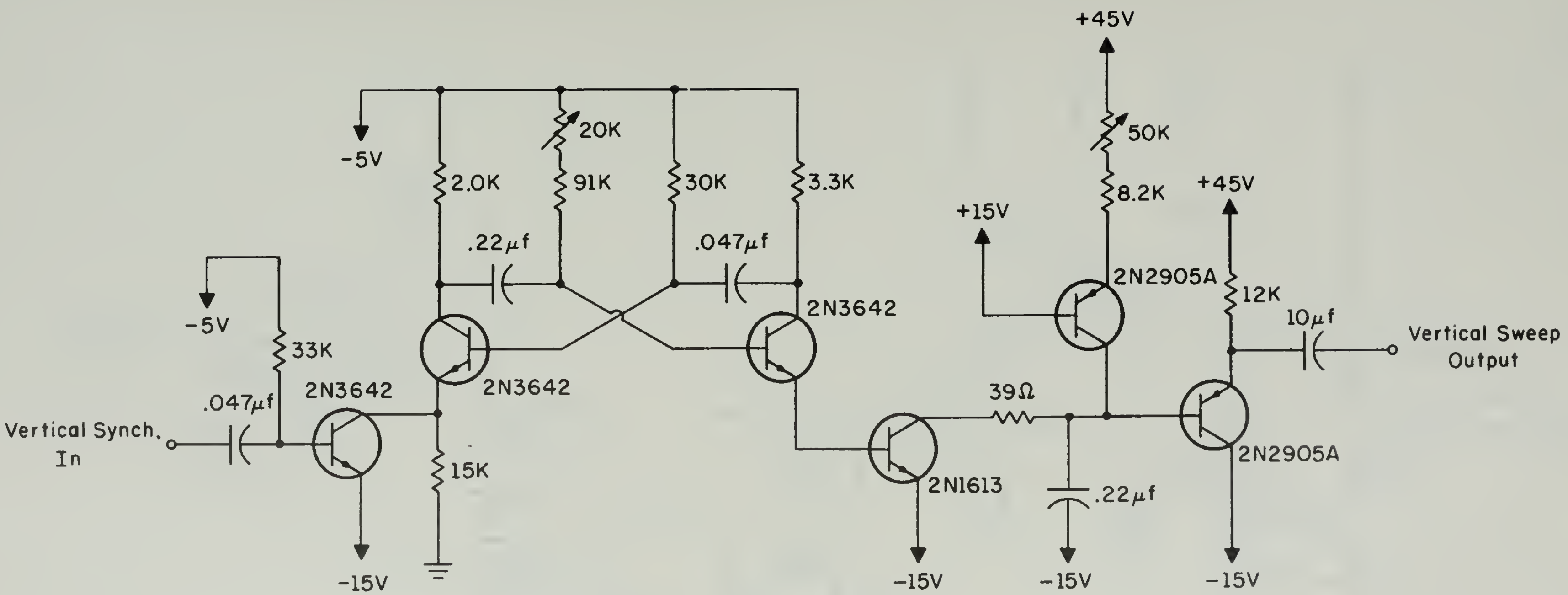


Figure 2. Vertical Sweep Circuit

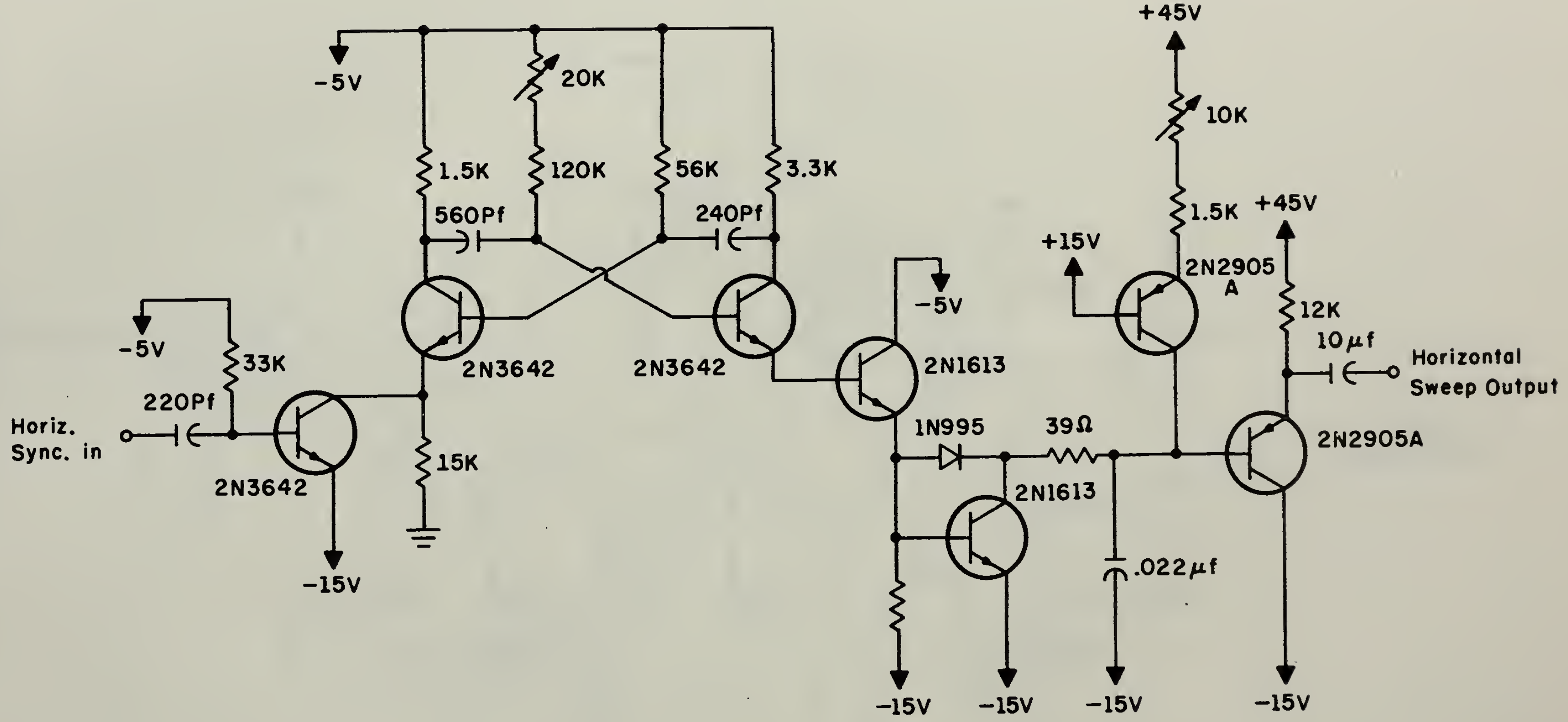


Figure 3. Horizontal Sweep Circuit

that similar circuits were used in Artrix.) Figure 4 shows the Video Amplifier which feeds a video signal to the cathode for intensity modulation.

Since a flying spot scanner is now being used in place of a pattern generator, it is necessary to raise the voltage across the CRT from 3 to 5kV. Preliminary tests indicate this will solve the intensity problem.

If further problems are not encountered, future work will be directed at determining an optimum decoding technique free of pattern orientation.

Dick Blandford

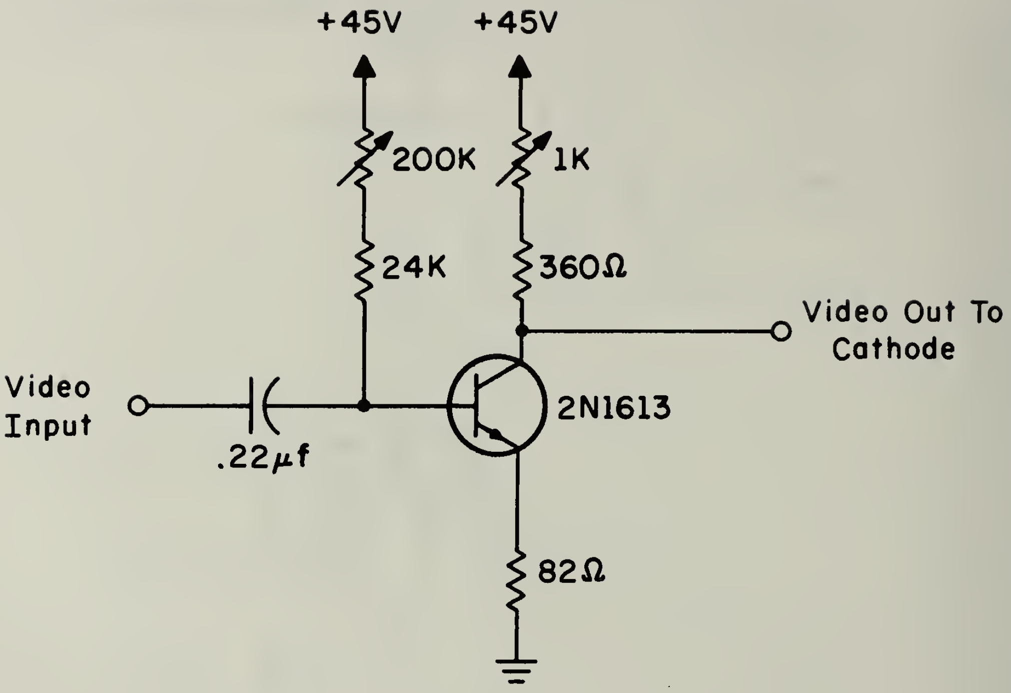


Figure 4. Video Amplifier

2.10 RASER (Project No. 29)

2.10.1 Summary

A method of expanding the display by a factor of two or four has been devised, using shift registers and data selectors. Full details will be given in the next report.

Tak Katoh

2.11 Stereomatrix (Project No. 30)

2.11.1 Observer Position Detector

The encoder disc code was programmed and drawn up and, after a 4 to 1 photoreduction, was etched on a .005" thick sheet of stainless steel. The shaft mount and mirror platform for the disc were also designed and fabricated. This subassembly awaits the reader mount for final testing.

An experimental stand was designed and built to test the reader electronics during encoder disc fabrication. Tests showed that the circuitry worked correctly. But they also indicated that the optic fiber diameter should be reduced in order to improve the resolution of adjacent binary numbers on the disc.

The encoder disc reader mounts have been designed and will be built next quarter.

Work to be completed during the next quarter will include design and construction of the detector cover and mount, fabrication of the necessary printed circuit boards for digital and analog circuitry, and interconnection of the subassemblies.

Chuck Pirnat

2.11.2 Coefficient Generator

All circuits except the master sequencer of the coefficient generator have been designed. The overall block diagram is shown in Figure 1. The 10-bit digital multiptier and multiplier control circuits are shown in Figures 2 and 3. Both of these have been tested. Completed during the past quarter are: (i) the translation control circuit, Figure 4, which enables the three-dimensional figure to be moved along x, y or z axes; (ii) the summing circuit, Figure 5, which adds/subtracts the sine product to/from the cosine product; (iii) the sign detector, Figure 6; (iv) the coordinate selector, Figure 7, for the

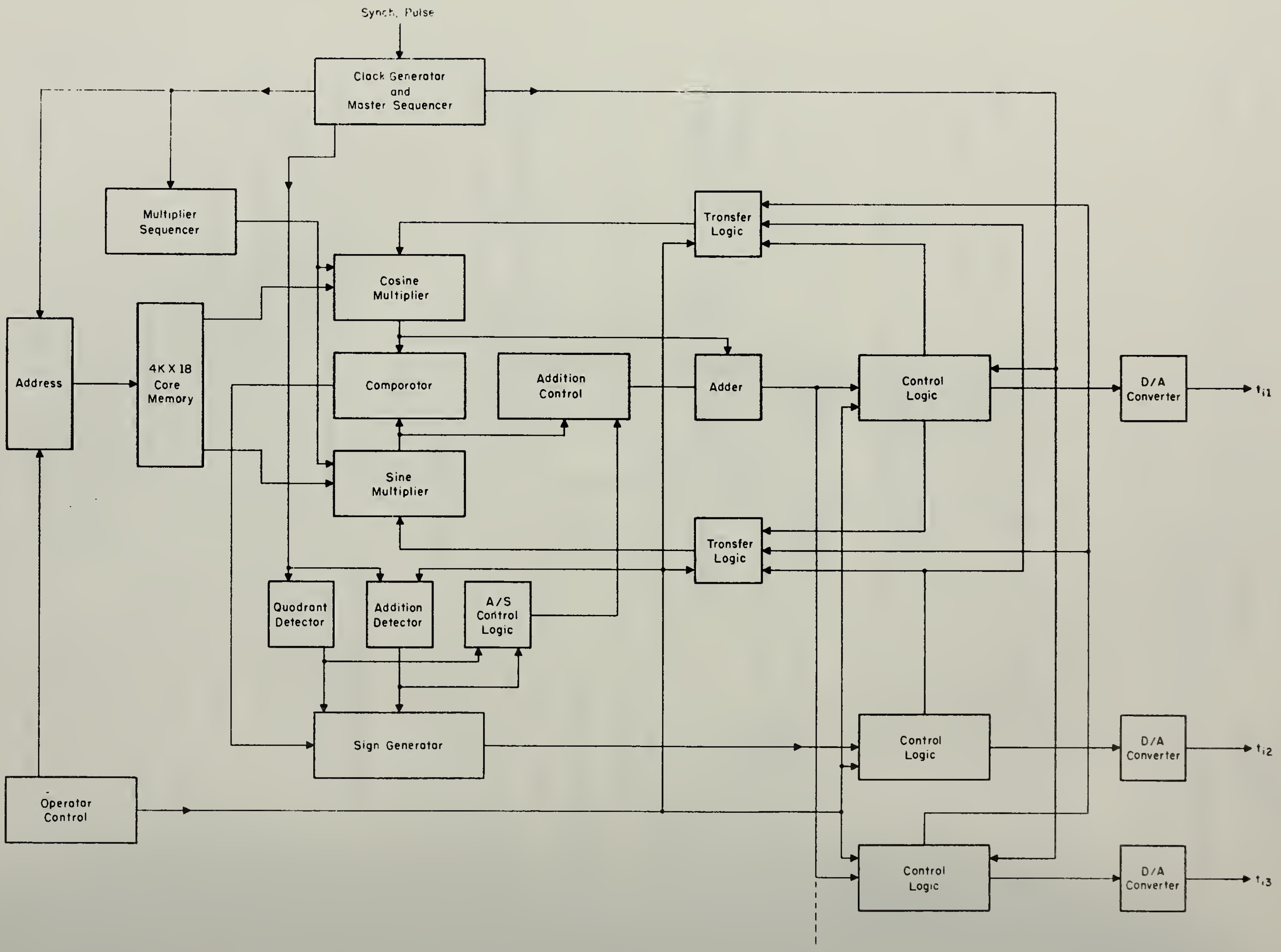


Figure 1. Stereomatrix Coefficient Generator

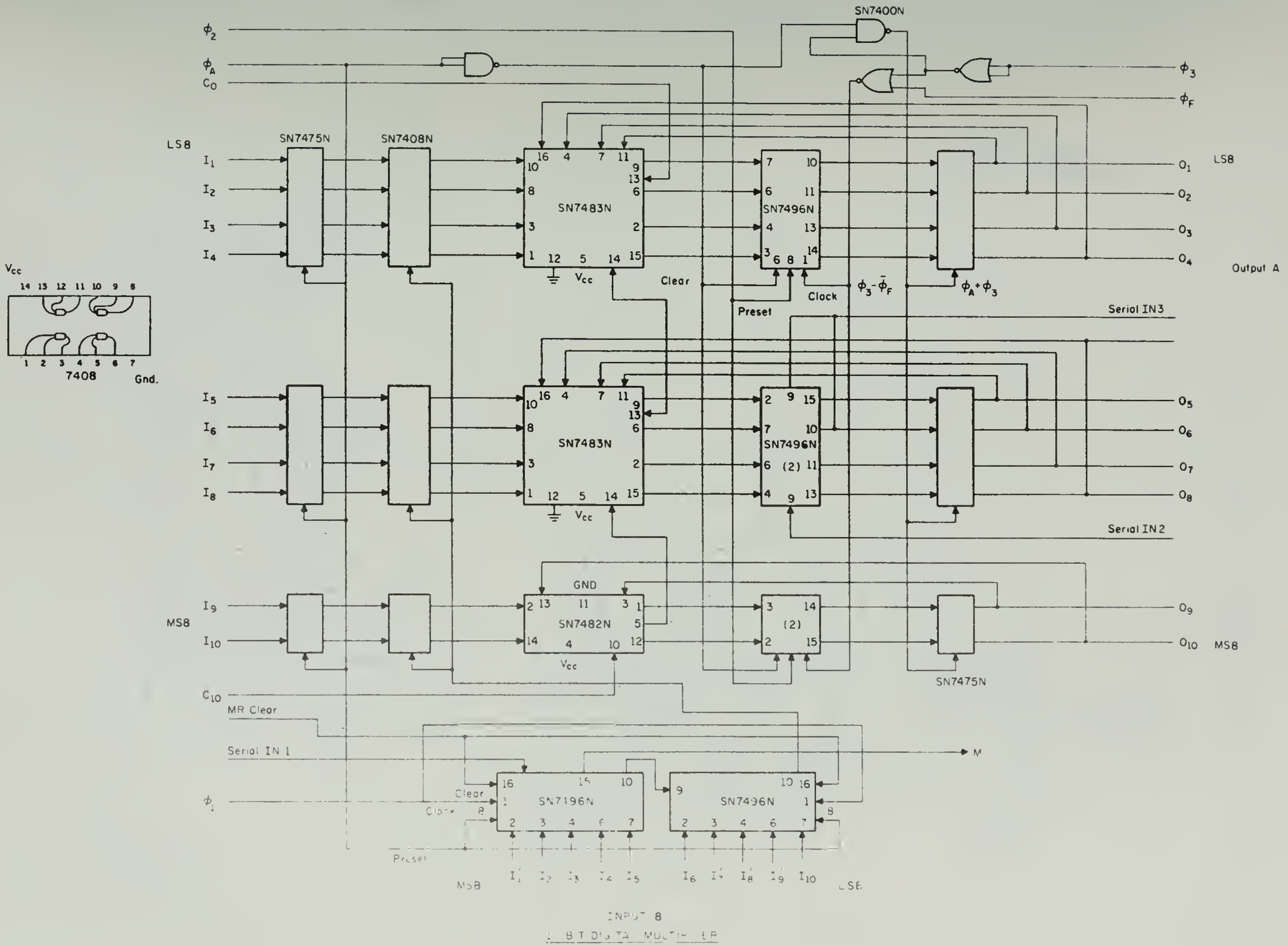


Figure 2. 10-Bit Digital Multiplier

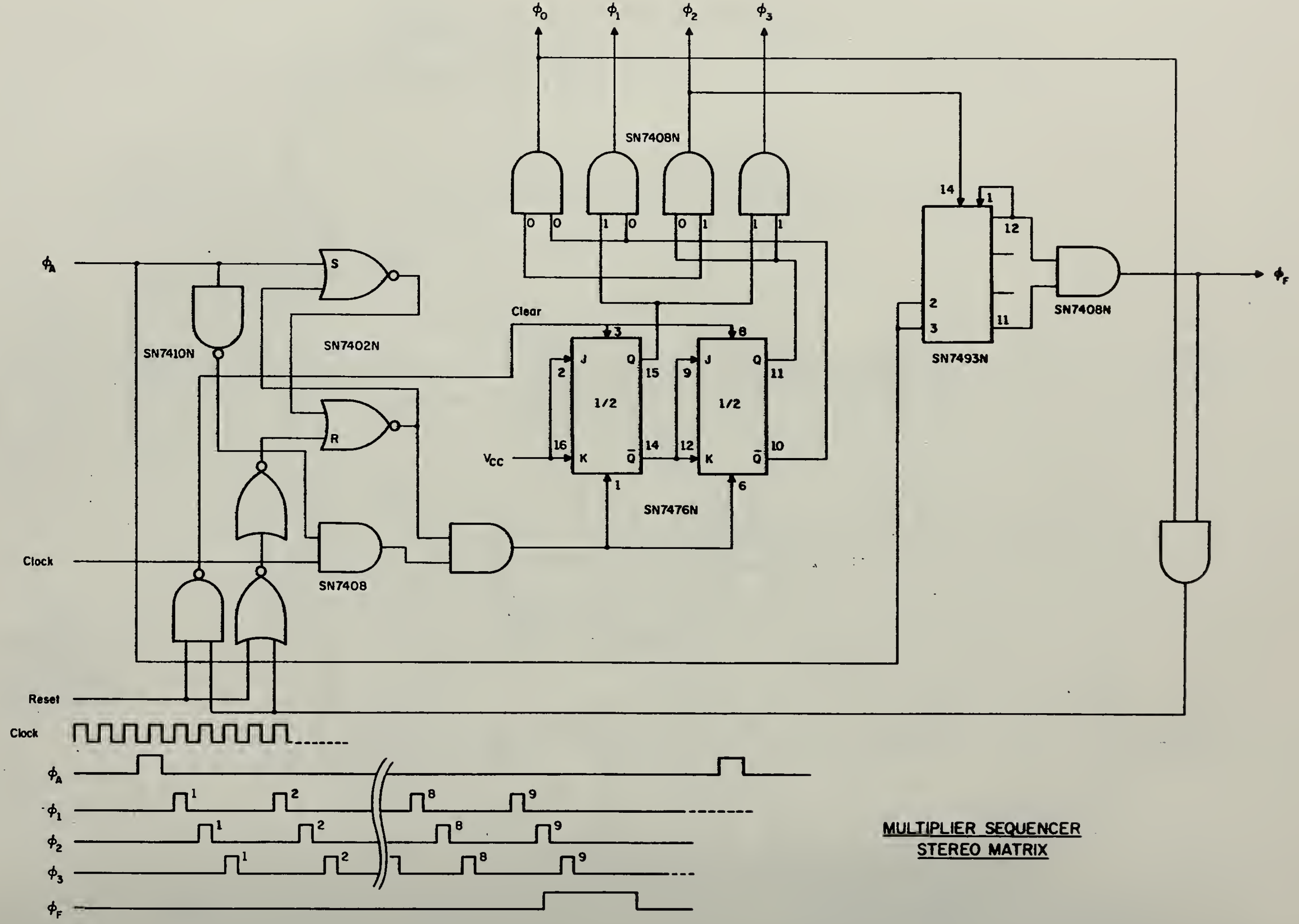


Figure 3. Multiplier Sequencer

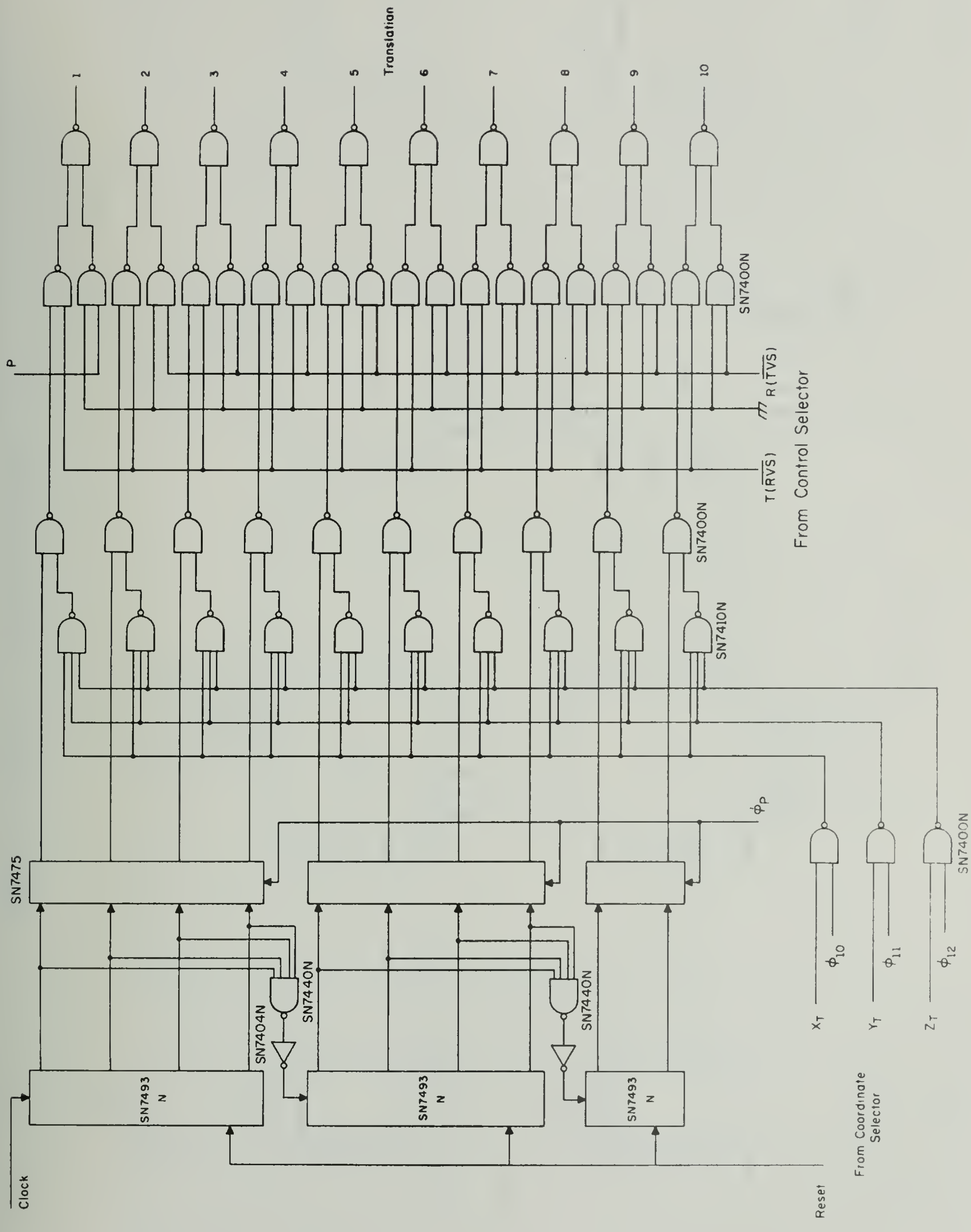


Figure 4. Translation Control (A)

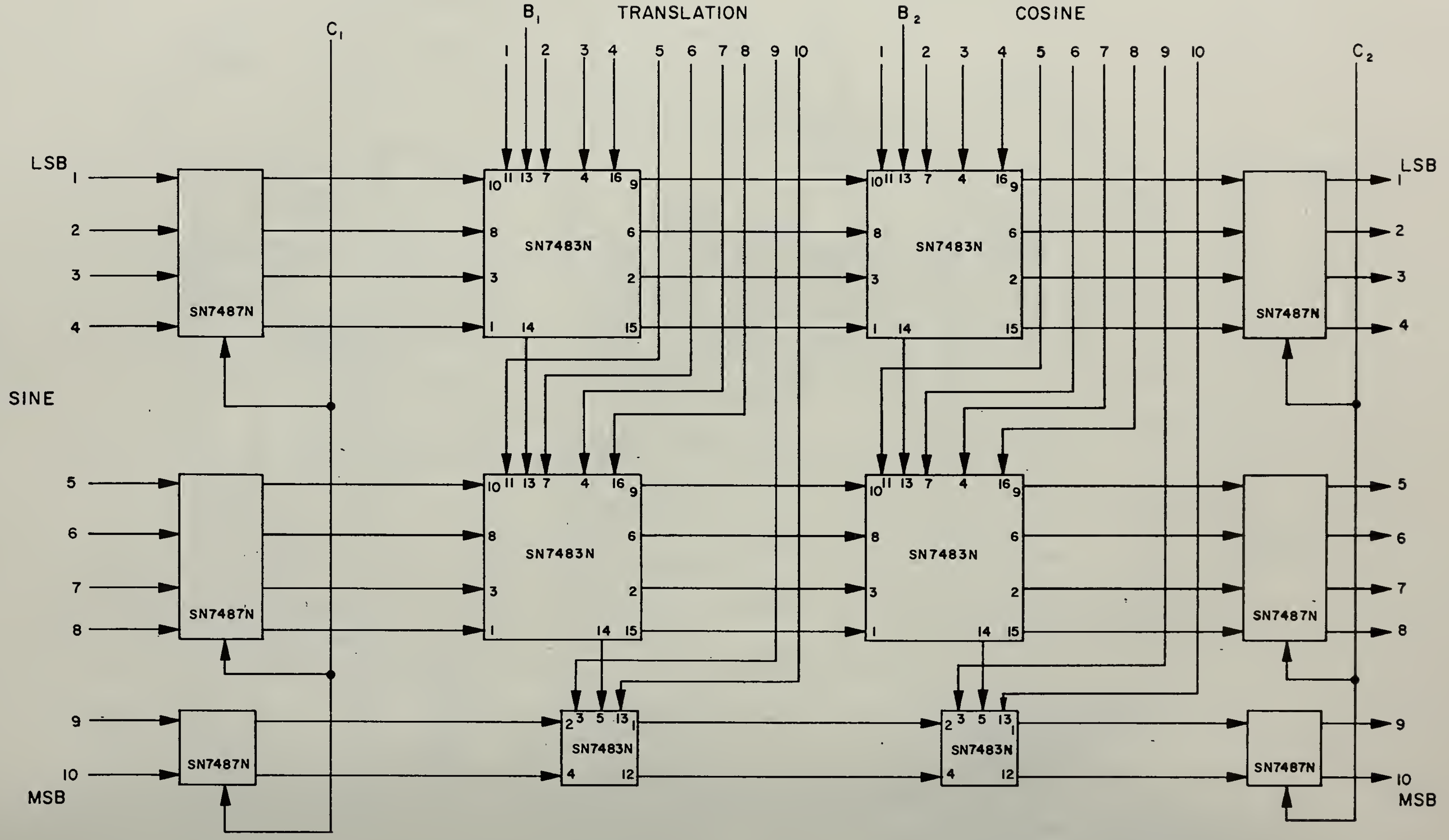


Figure 4 (Cont.). Translation Control (B)

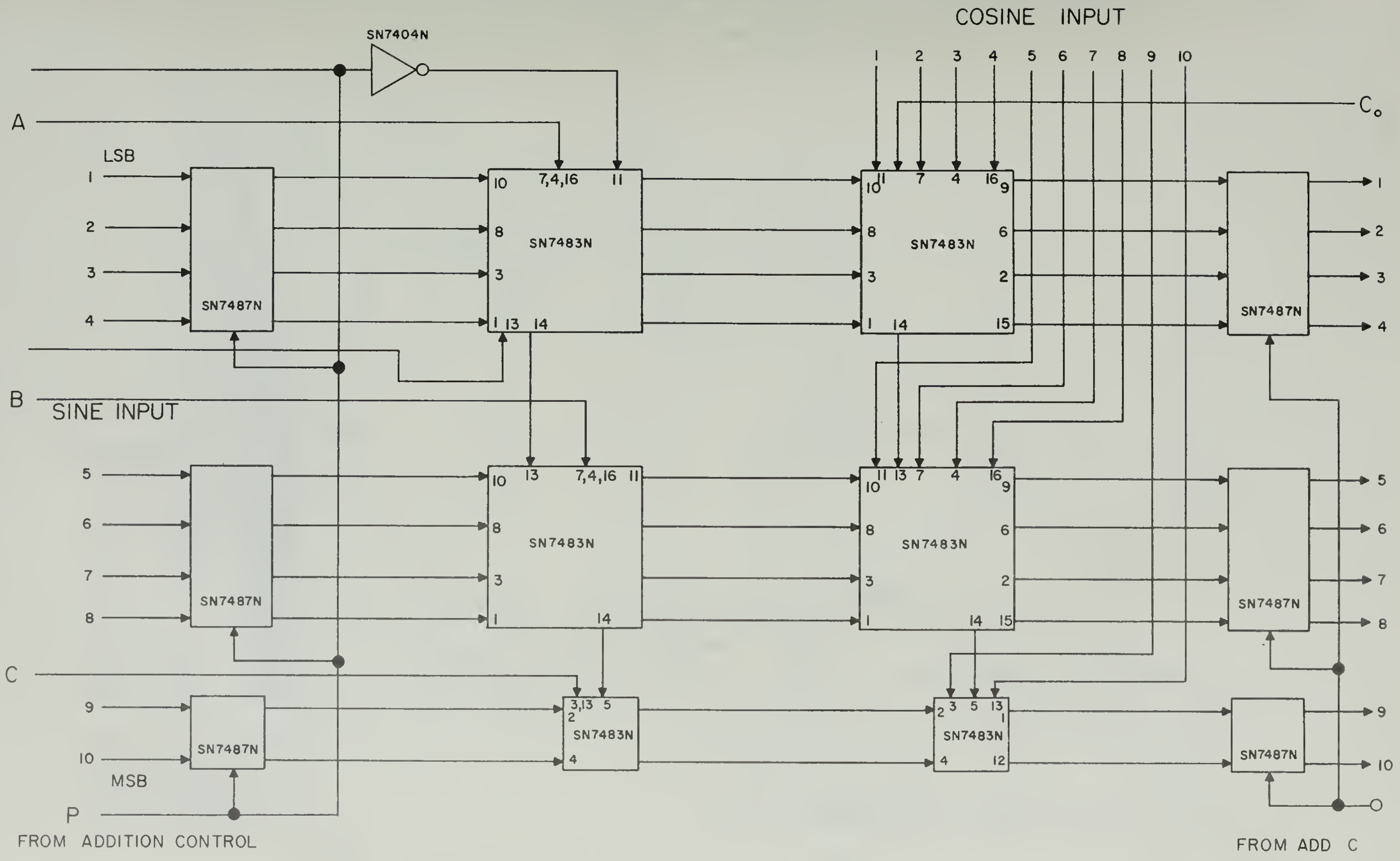


Figure 5. Summing Unit

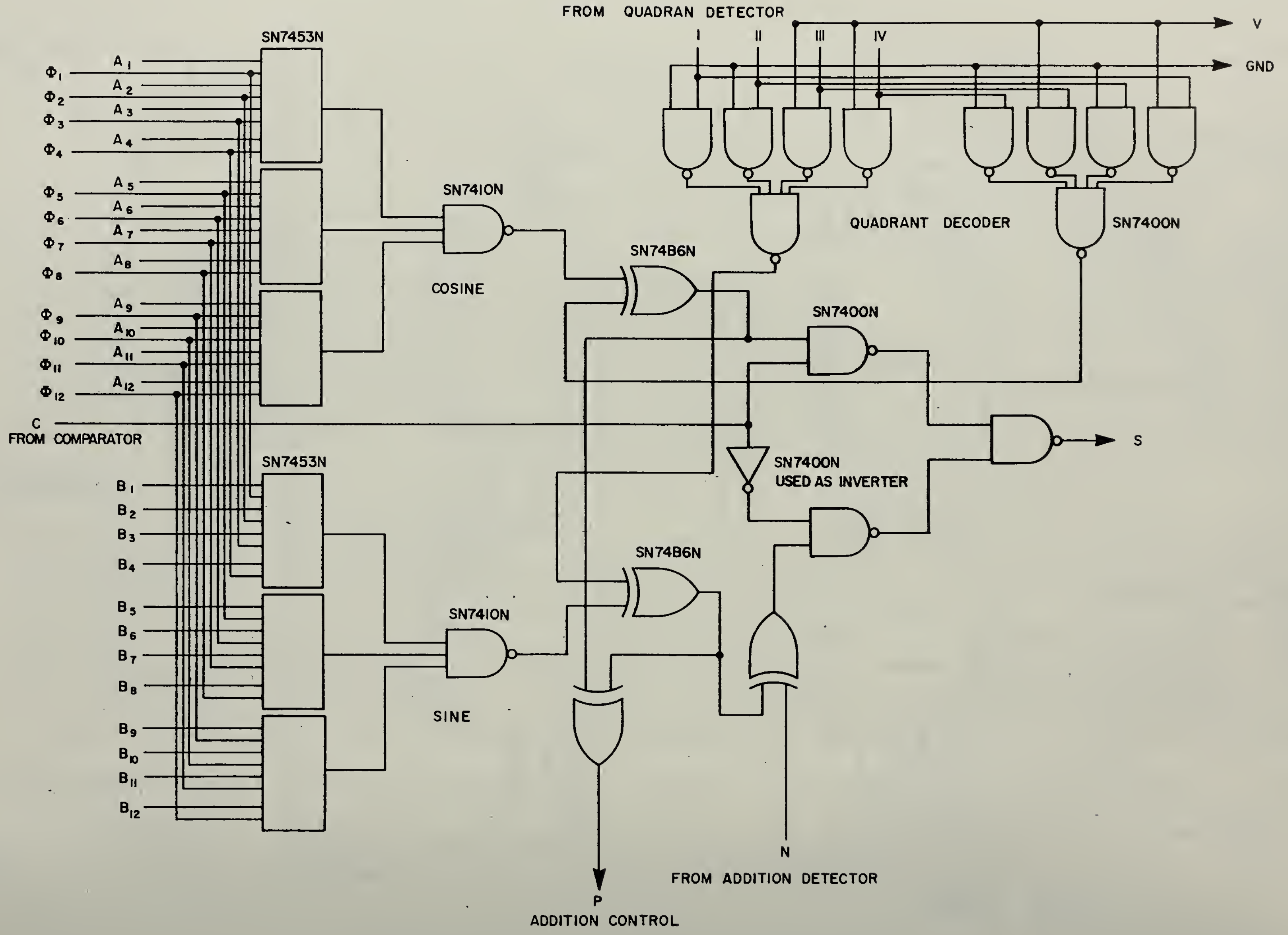


Figure 6. Coefficient Sign Generator

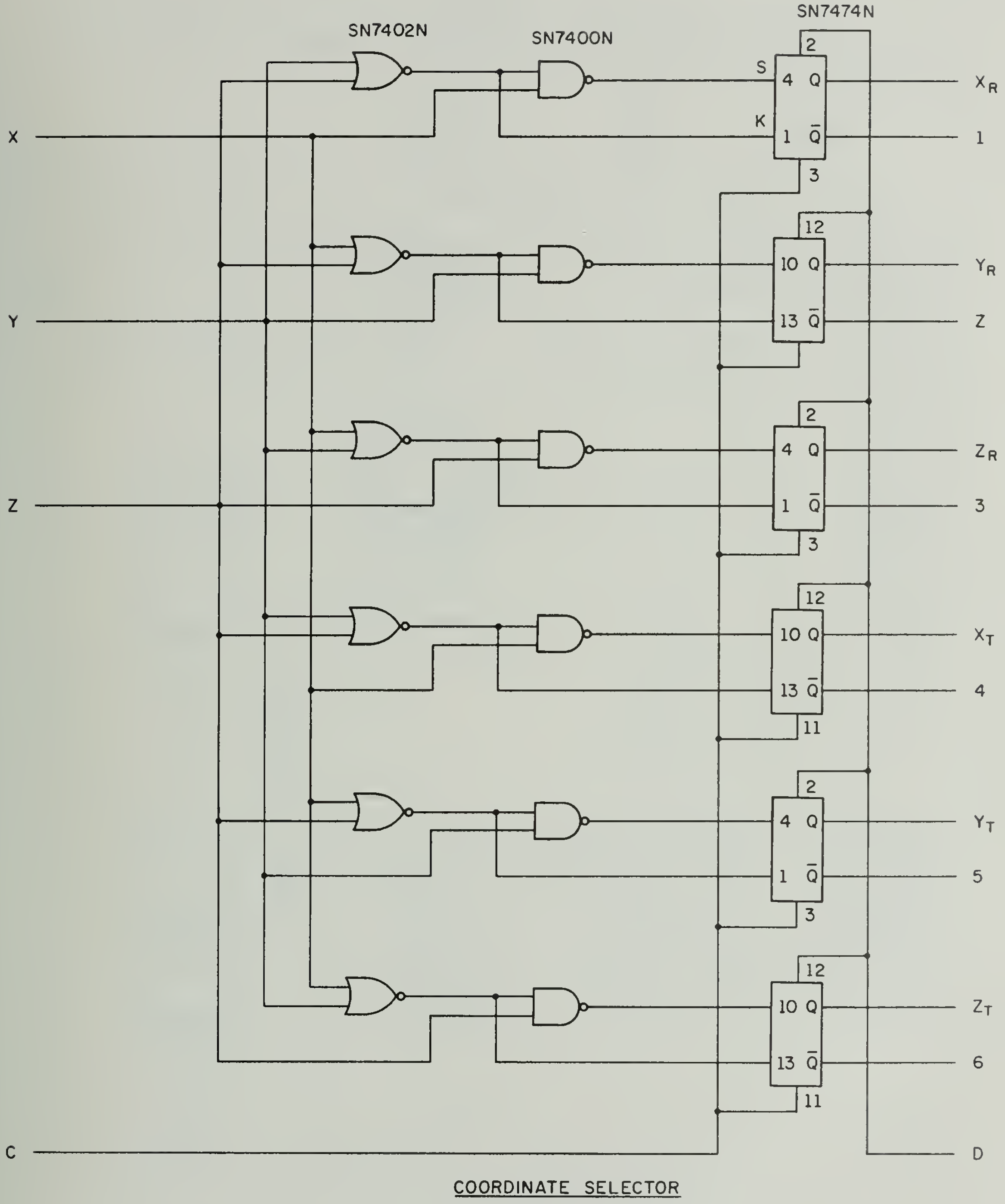


Figure 7. Coordinate Selector

operator's control panel. All circuits are being tested and are ready for fabrication. The most complicated part of the chassis wire wrapping which is the computing control section, has been completed.

Things remaining to be done are (i) memory loading; (ii) master sequencer designing; and (iii) wire wrapping of the rest of the system.

2.11.2 Cursor

All of the cursor circuits have been designed and more than half have been built. The system block diagram is shown in Figure 8. The most complicated subsystem is the analog matrix universe transformer which operates on a 4 × 4 matrix with $t_{41} = t_{42} = t_{43} = 0$ and $t_{44} = 1$; this is shown in Figure 9. The coincidence detector and circle generator circuits are shown in Figures 10 and 11.

Dick Cheng

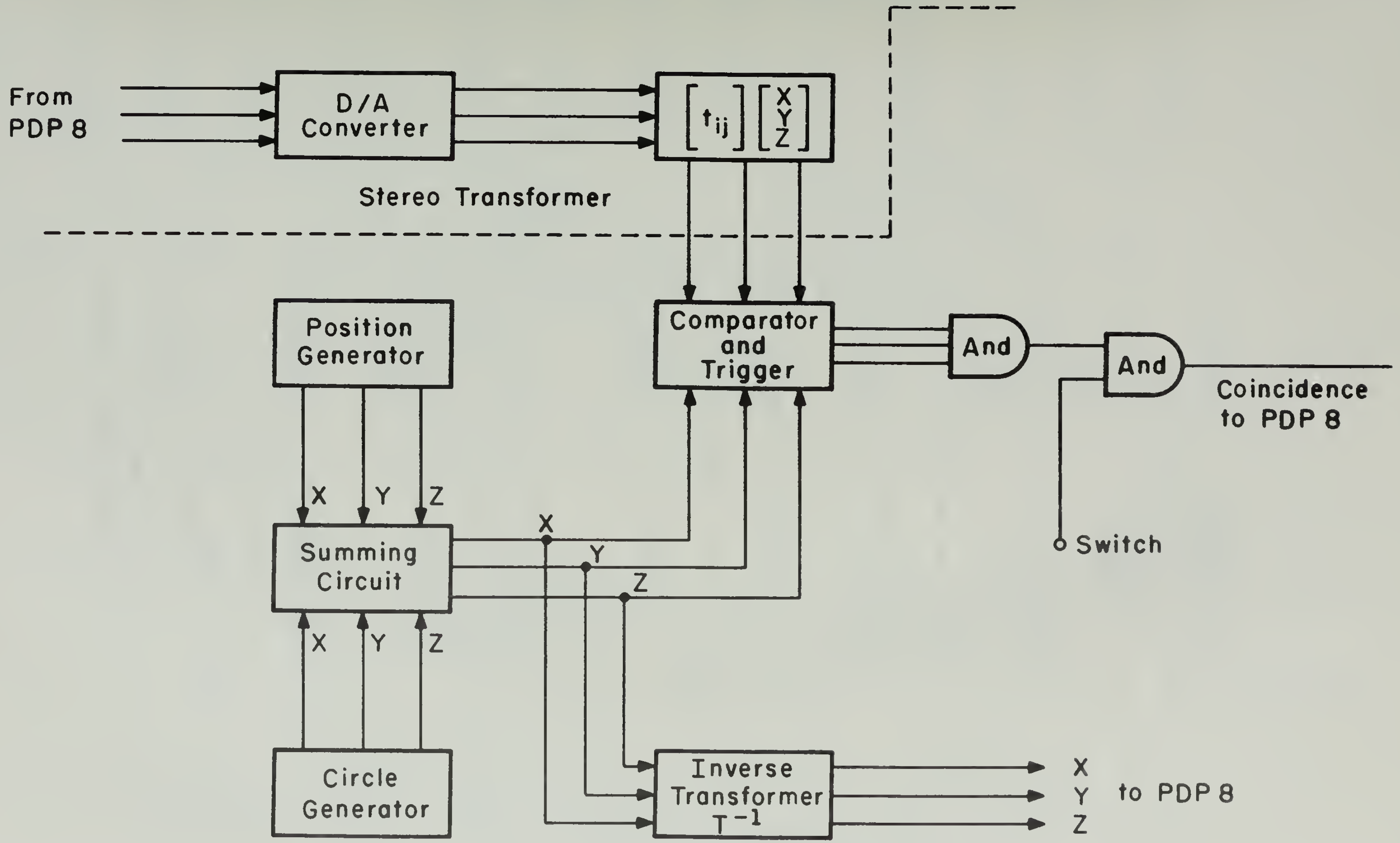


Figure 8. Cursor Block Diagram

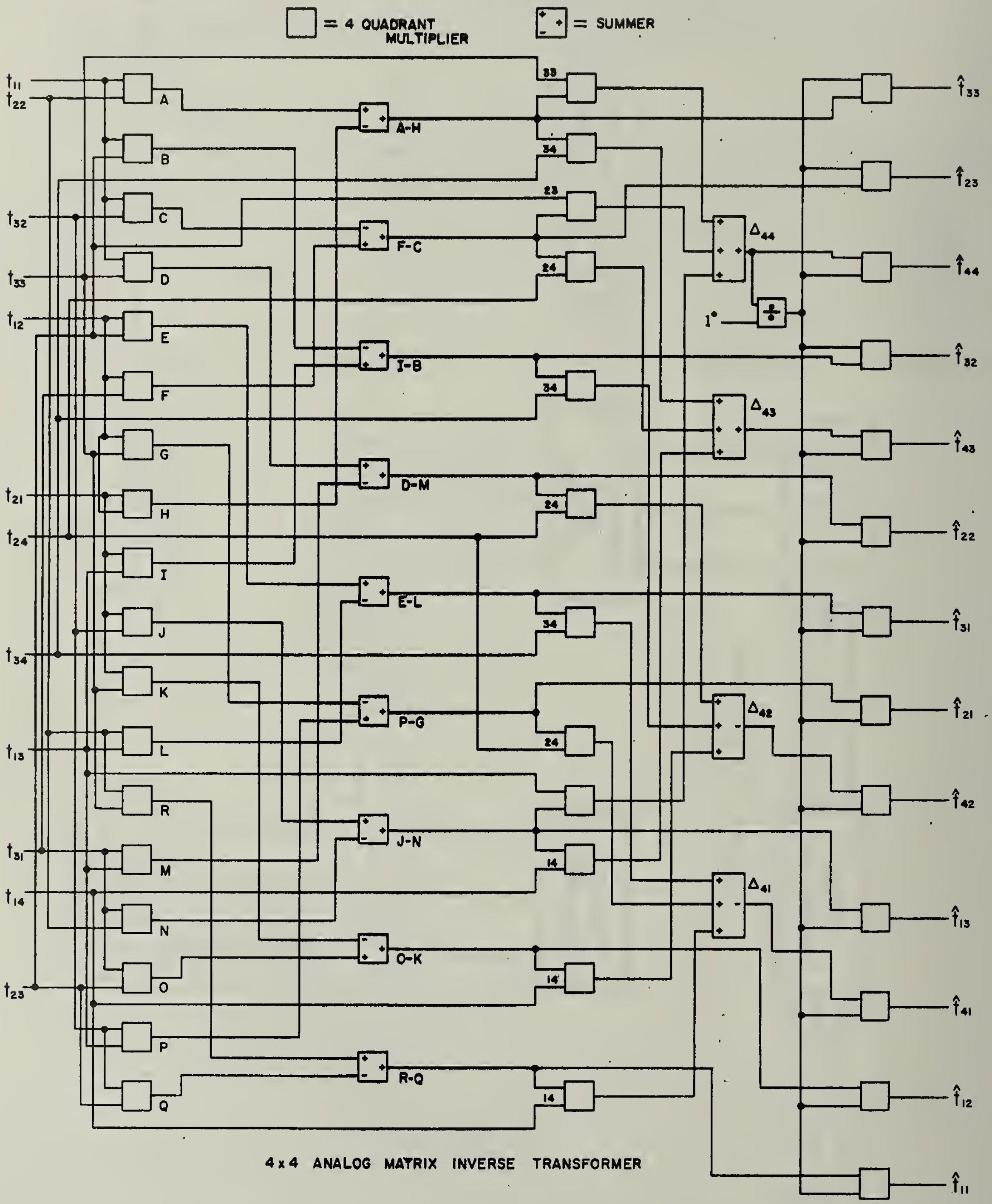
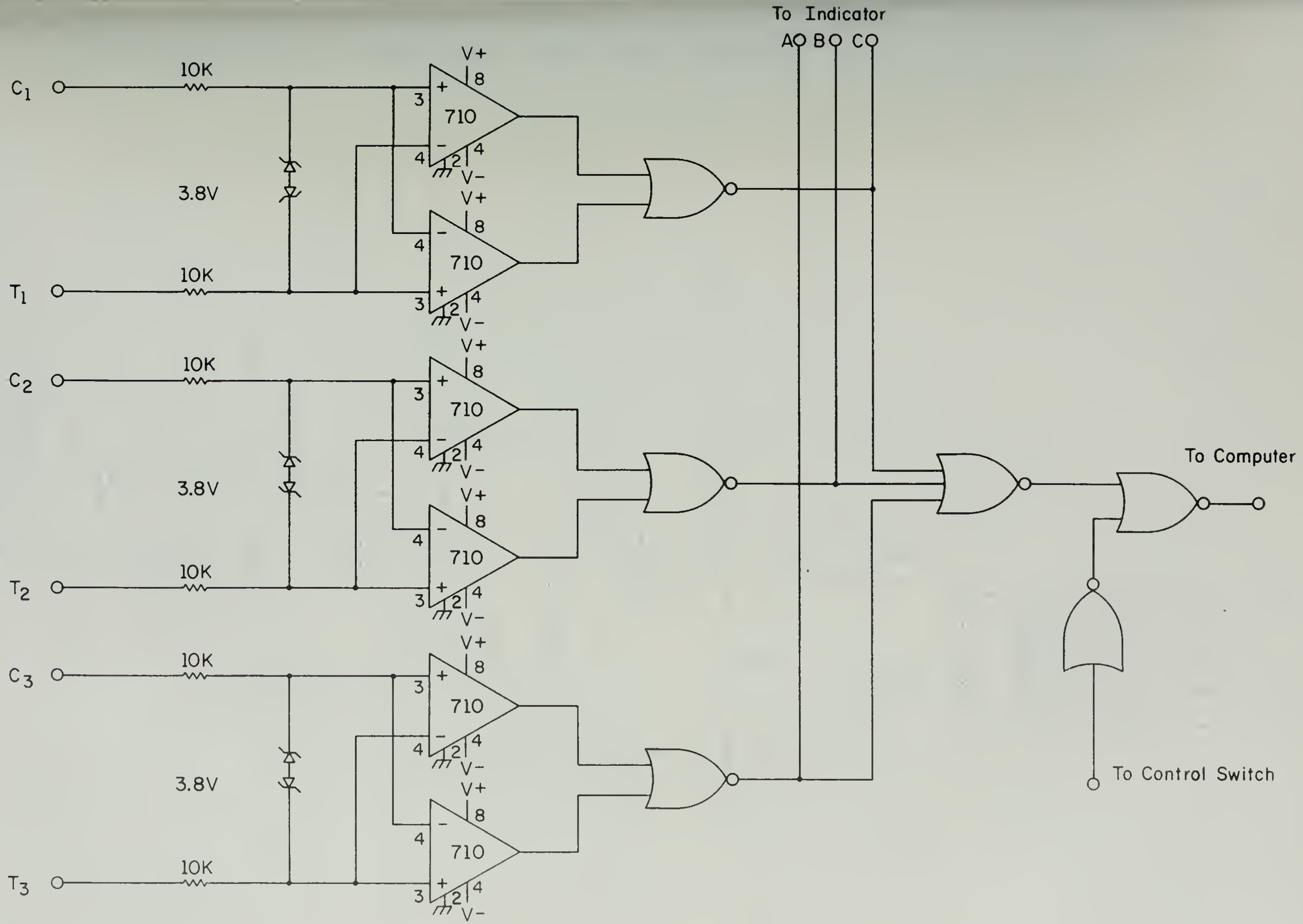


Figure 9. Analog Matrix Inverse Transformer



COINCIDENCE DETECTOR
CURSOR, STEREOMATRIX

Figure 10. Cursor Coincidence Detector

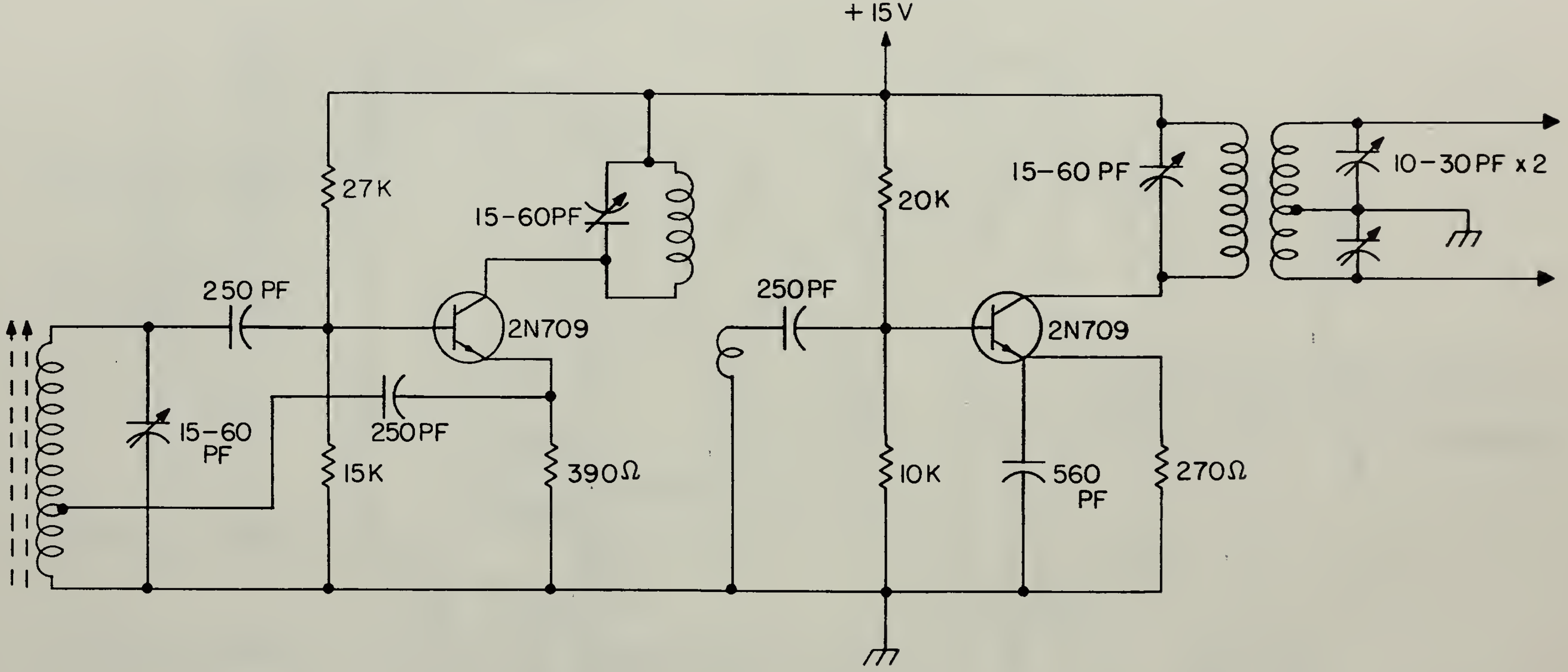


Figure 11. Cursor Circle Generator

2.12 PAGAN (Project No. 32)

2.12.1 Polyhedron Generation

As described in the last quarterly report, PAGAN will generate polygons in cylindrical coordinates by appropriately varying the radius, R, as a function of the angle Θ. Figure 1 is a block diagram of polygon generation. Also discussed in the last quarterly report was the desirability of digital synchronization among internal waveforms. Therefore, Θ and Φ are derived digitally and converted to analog signals. A polar to rectangular transformation, computed using analog modules, yields x and y outputs.

If Z is incremented each time the polygon is outlined, a prism results. Furthermore, if r, the radius of the inscribed circle, is decremented when Z is incremented, then a pyramid is formed. Figure 2 shows the creation of solids having a regular polygon cross section. The proper increments and limits for each figure are determined by the digital Pattern Programming unit.

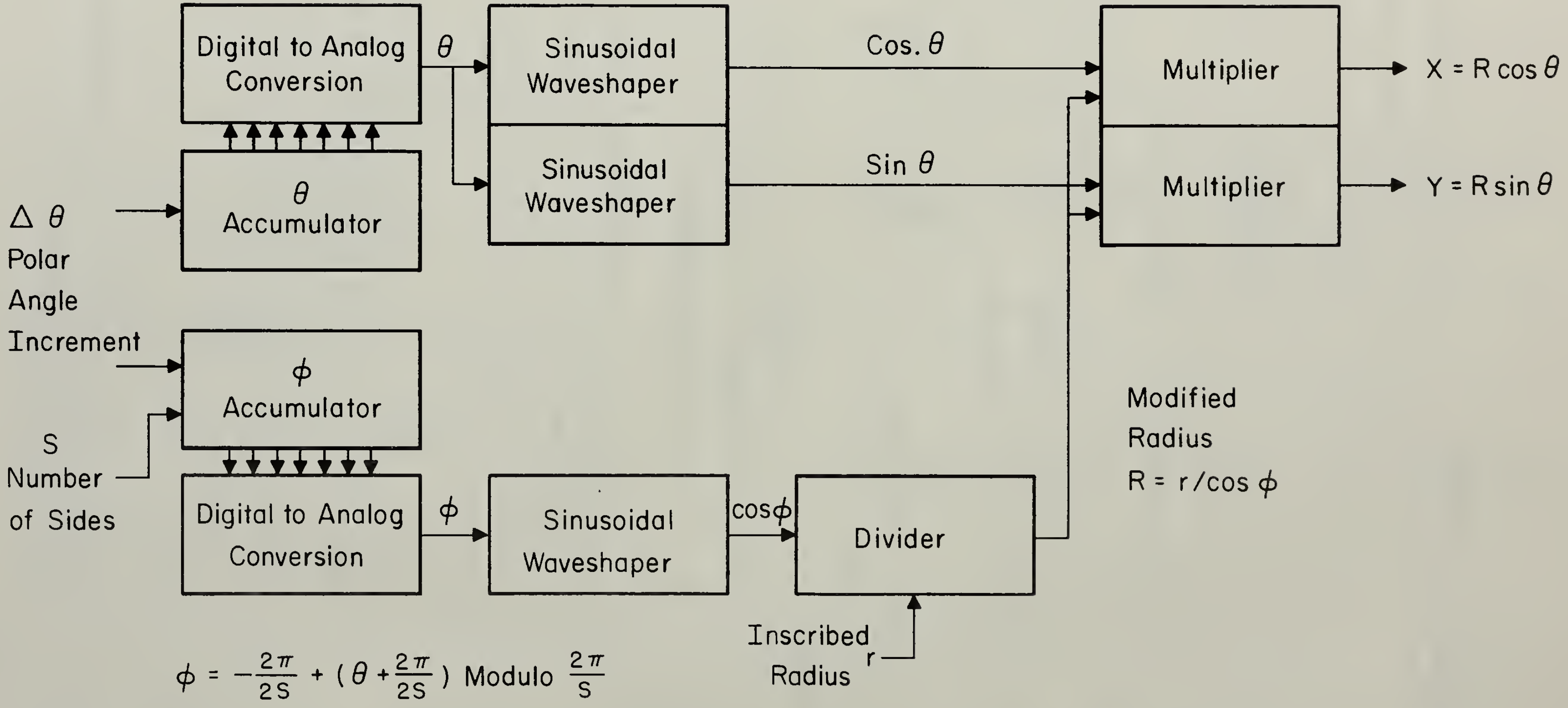
2.12.2 Circuit Design

At the end of the fourth quarter, a preliminary breadboarding of PAGAN was being used to evaluate D/A converters and operational amplifiers. During this past quarter, those items were ordered and have been delivered.

Also from the preliminary circuitry, three cards, shown in Figures 3, 4, 5 were submitted for printed circuit layout. Samples of the cards have recently been fabricated. These boards, as well as a couple of Stereomatrix designed cards, are used in PAGAN's polygon generation.

The multipliers and divider needed were ordered during the quarter. The divider module is actually a multiplier placed in the feedback loop of an operational amplifier circuit. Thus, frequency response and accuracy are inversely proportional to denominator magnitude. The divider module was chosen after considering the range of the divisor, cos Φ.

In the past quarter, the actual construction of PAGAN began. Card racks and power supplies were acquired and assembled into a cabinet. Wire lists have been drawn up and physical wiring of the back panel is about to begin.



Generation of a Regular Polygon of "S" Sides

Figure 1. Generation of a Regular Polygon of S Sides

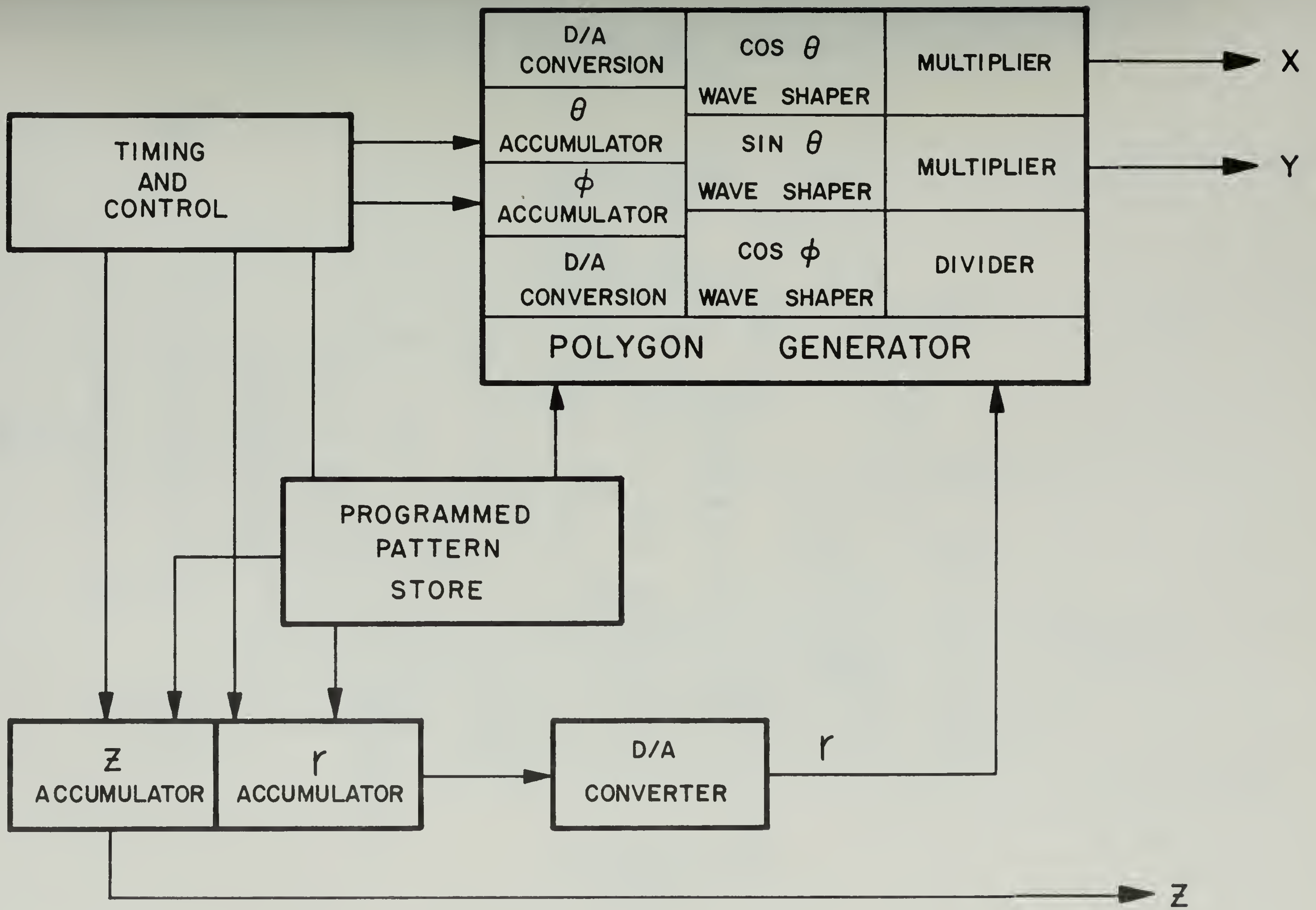


Figure 2. Generation of Polyhedral Surfaces

1469 - 484

4 Bit Adder-Subtractor

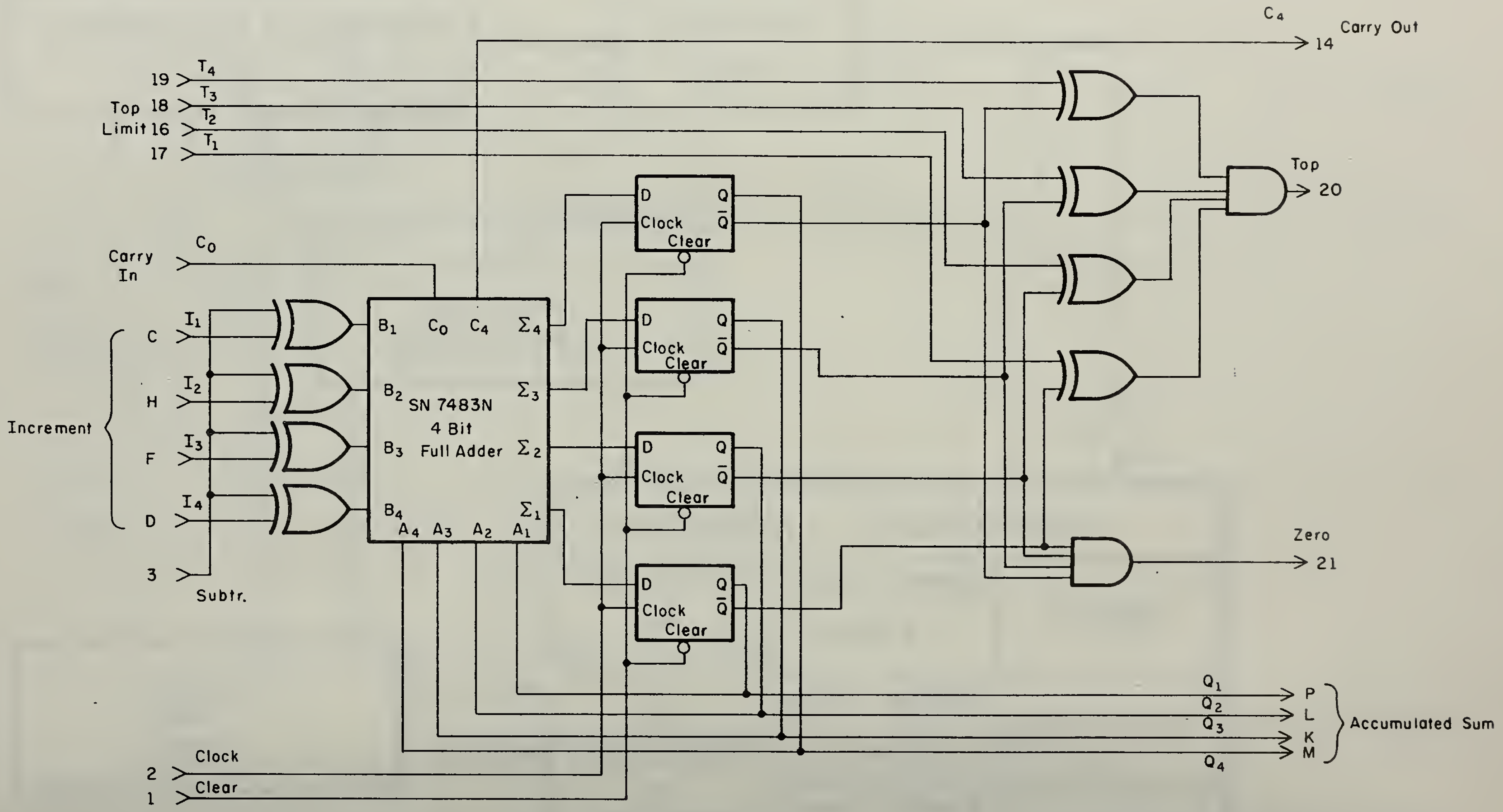


Figure 3. PAGAN 4-Bit Adder - Subtractor

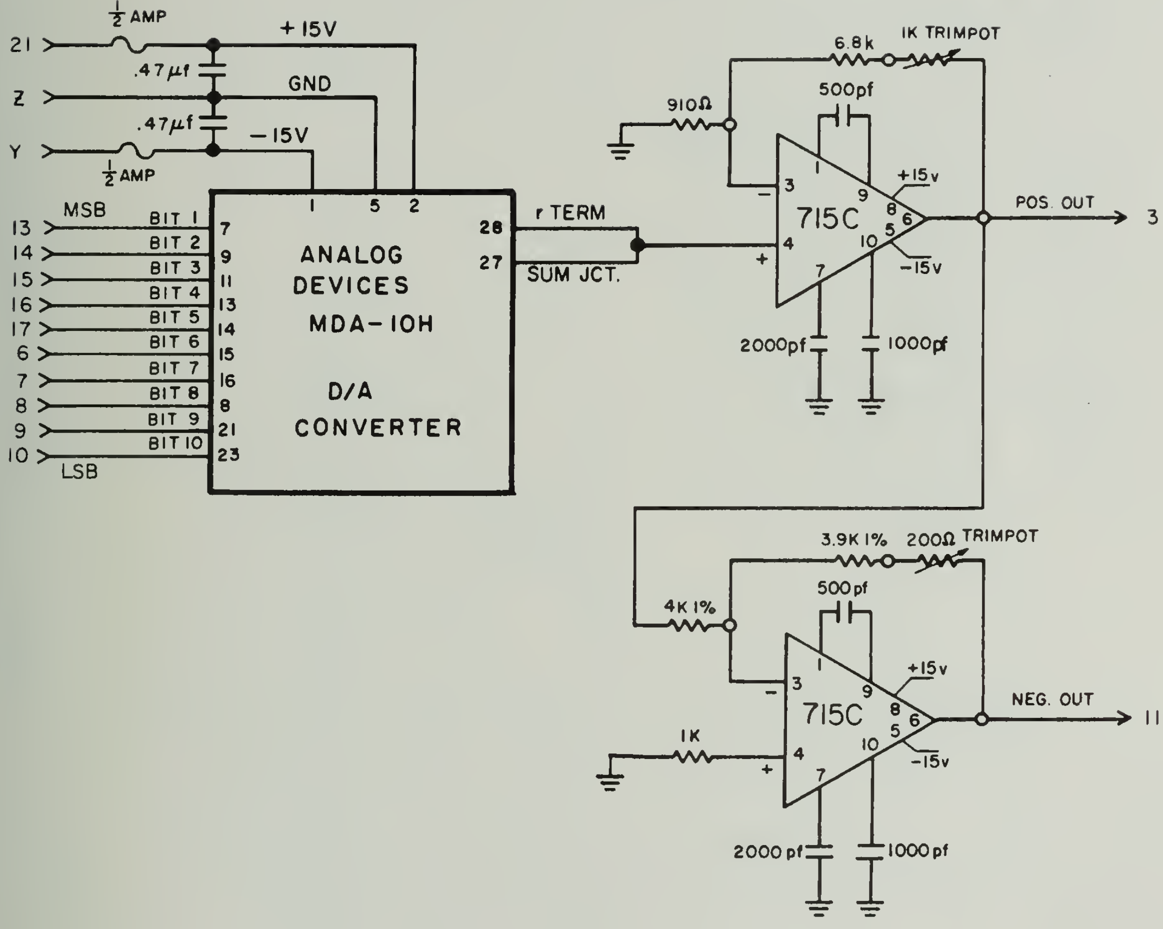


Figure 4. PAGAN Θ, Φ D/A Conversion

Generation of Polyhedral Surfaces

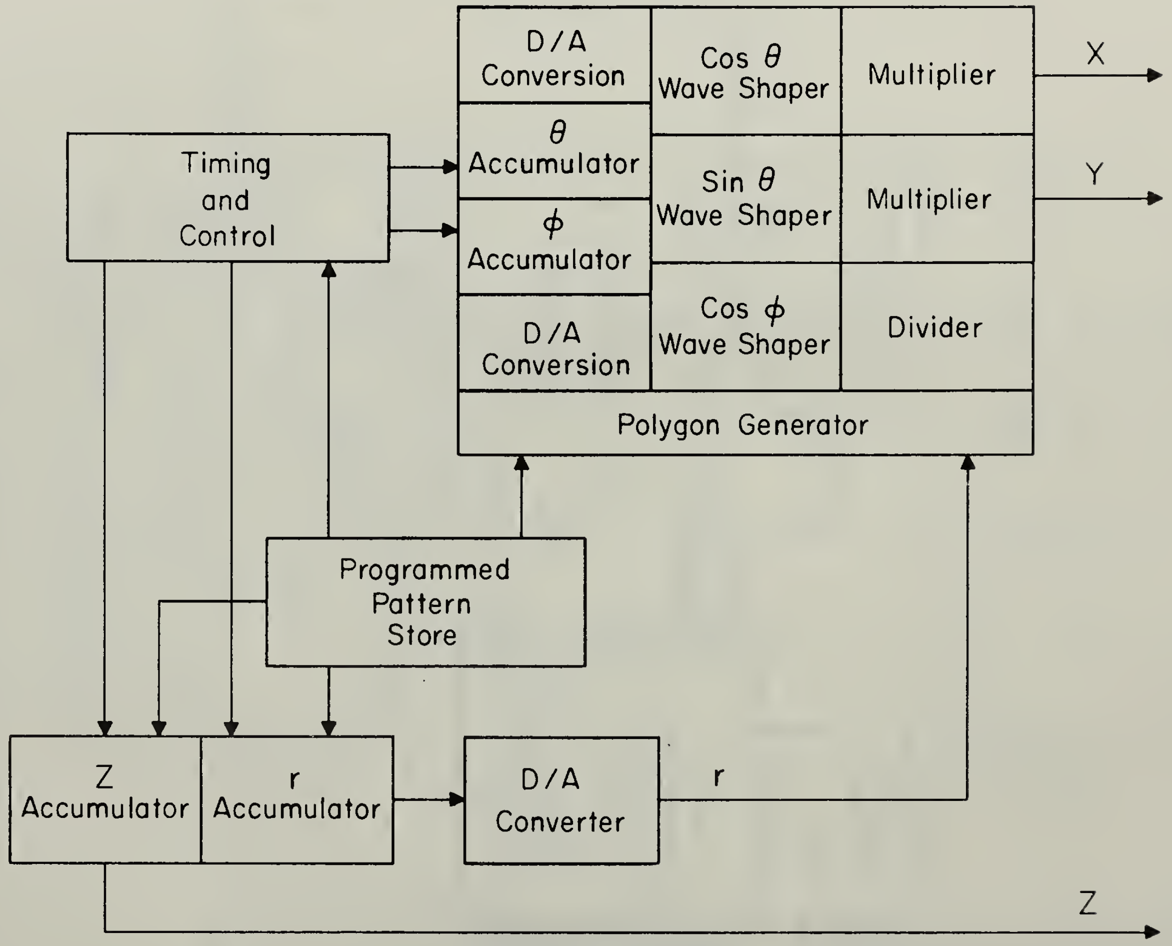


Figure 5. Sinusoidal Transconductor

2.12.3 Point Spacing

With accurate analog components and equal angle increments, there will be uniform point spacing on a circular cross section. However, as the circle is shaped into a polygon, the point spacing will no longer be truly uniform. Referring to Figure 6, if the angle increments remain equal the point spacing will vary corresponding to rtan Φ. If the spacing is to remain uniform, the angle must be incremented in proportion to the arctan of that spacing.

Polar coordinates were decided upon because of the simplicity and generality in creating figures. Since an objective of the PAGAN project is building a relatively simple and inexpensive pattern generator, point spacing compensation will not be attempted. Although the nonuniform spacing may be evident in a triangle, it is inconspicuous in a polygon of five or more sides.

Richard L. Partridge

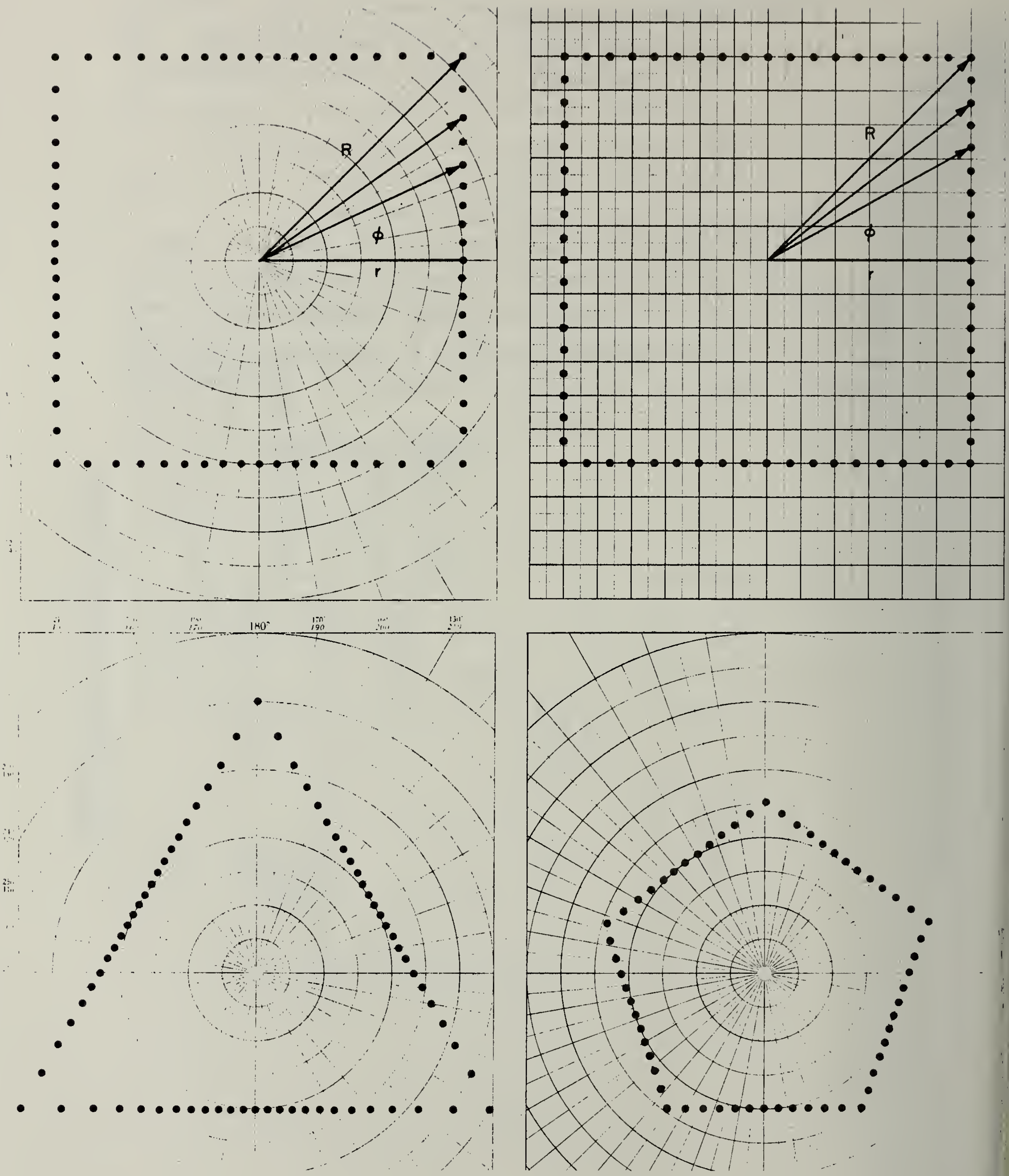


Figure 6. Point Spacing

2.13 Scantrix (Project No. 35)

More sophisticated methods of fabrication have made light emitting diodes cheaper and, if this trend continues, it is very likely that in a very short period of time there might be available diodes which cost about 30 cents each. This has encouraged us to give up the idea of a rotating mirror to scan the whole line of frozen TV signal and instead build a frame of 128 × 128 light emitting diodes with two alternating registers driving each line of diodes in sequence. One apparent advantage of this approach is that it eliminates the bulky and complicated structure of a four-foot long rotating mirror system while at the same time makes a flat display possible.

About six diodes of different makes and different characteristics are now under study. It is hoped that some circuits will be designed before September when the financial situation gets better and the project will no longer be delayed.

Sik Yuen

Form AEC-427
(6/68)
AECM 3201

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY–TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO. COO-1469-0182	2. TITLE QUARTERLY PROGRESS REPORT - JANUARY-MARCH, 1971

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report

□ b. Conference paper not to be published in a journal:

Title of conference ______

Date of conference ______

Exact location of conference ______

Sponsoring organization ______

□ c. Other (Specify) ______

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.

□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.

□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

Professor W. J. Poppelbaum
Principla Investigator
Hardware Research Group

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature	Date March 31, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.

□ b. Report has been sent to responsible AEC patent group for clearance.

□ c. Patent clearance not required.

3. SOFTWARE SYSTEMS RESEARCH

All of the packages necessary for the transient analysis of arbitrary networks have been completed this quarter and work has just begun on linking them together.

A multiplexor enabling up to eight storage tube displays to be driven by the PDP-8/I has been completed and is operational. The new disk controller is now operational.

Publications this quarter include:

Gear, C. W. "The Automatic Integration of Ordinary Differential Equations," Communications of the ACM, 14, #3 (1971), pp. 185-190, along with submission to the Algorithm section of CACM).

Gear, C. W. "The Simultaneous Numerical Solution of Differential-Algebraic Equations," IEEE Transactions on Circuit Theory, TC-18, #1 (1971).

Gear, C. W. "A Graphical Search for Stiffly Stable Methods (with C. Dill)," Journal of the ACM, 18, #1 (1971), pp. 75-79.

Gear, C. W. "Transient and Steady State Numerical Solution of Differential-Algebraic Equations," IEEE Conference Proceedings, 1 (1971), pp. 384-386.

C. W. Gear, Professor
and Principal Investigator

3.1 Numerical Processes

The set of packages which performs the numerical and associated processing for transient analysis has been completed. They have been linked together, and linked to use the output from the compiler stage described in section 3.2.1. A rough flow of the whole system is shown in Figure 1.

The separate packages are discussed below.

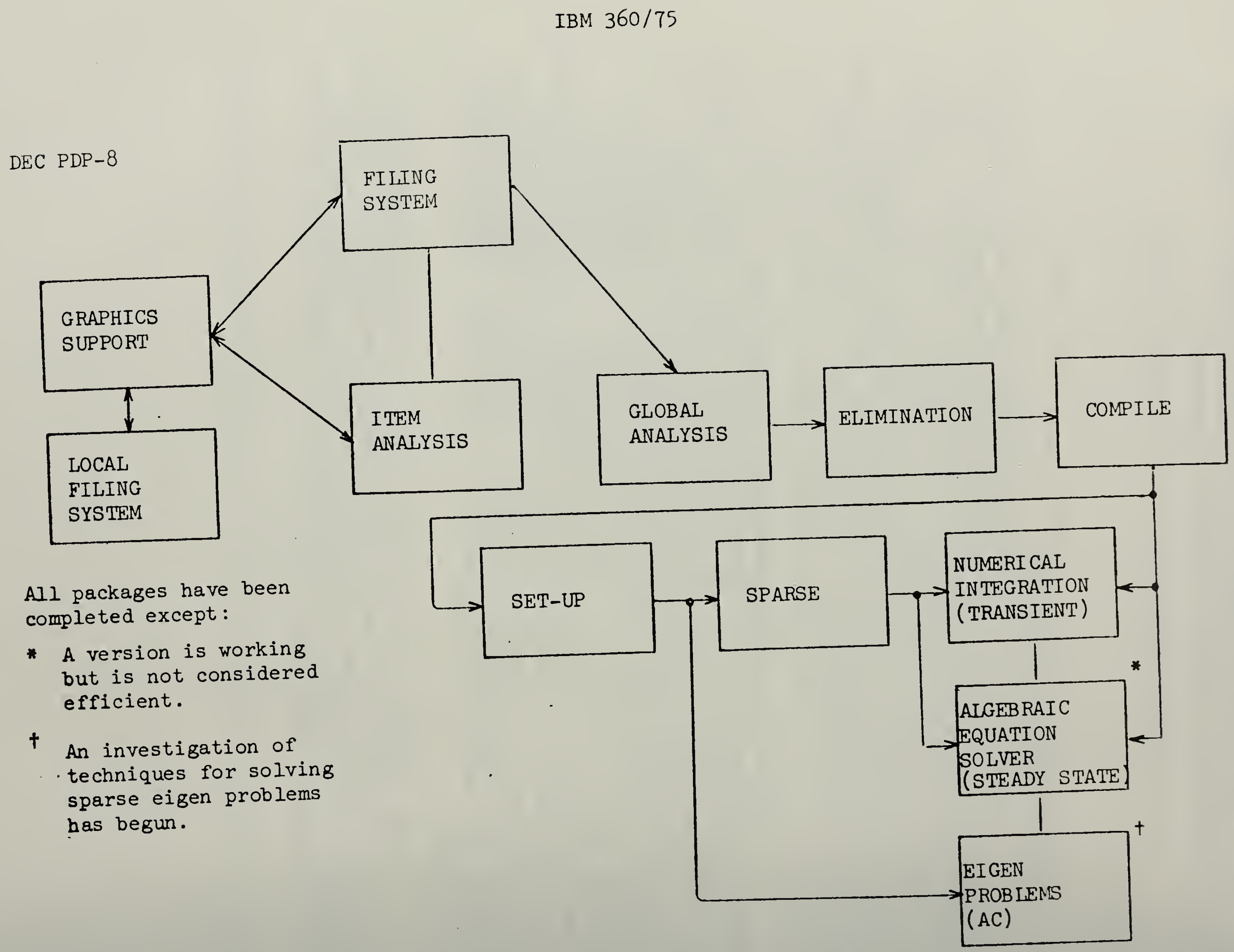
3.1.1 Sparse Matrix Inversion (J. Deogun, K. Ratliff)

The group of routines which generates code to invert sparse matrices has been combined with the programs (MAIN and DIFSUB) which integrate ordinary differential equations. The function of the sparse programs is to generate matrix inversion code for the Jacobian of a system of equations. This result is used by DIFSUB in the corrector step of the integration process. The combination of routines has run successfully with equations used previously for test purposes. Figure 2 gives a diagram of the relationship between these two sections.

Within the sparse programs an improvement in the method of storing matrix element values has been implemented. If a matrix element is not an integer, its value must be stored in a floating point array (PW) of arbitrary length. These elements are either fixed constants or symbolic variables. If sparse is generating code to do arithmetic involving both integers and nonintegers, it must first store the integers as floating point constants. Rather than using a new location for each of these, the first twenty locations of PW are initially set to contain the floating point form of all integers with absolute value $\leq$ 10 (zero excluded). When one of these values is required, a subroutine is called to determine which location contains the appropriate constant and thus a new location need not be used.

The section of the sparse program which selects the order of the pivots has been changed experimentally in an attempt to find an optimal pivot selection algorithm. The basis for comparison of algorithms is the number of instructions generated for subroutines MATMUL and MATINV.

The following methods have been tested with the sparse program.



All packages have been completed except:

* A version is working but is not considered efficient.

† An investigation of techniques for solving sparse eigen problems has begun.

Figure 1

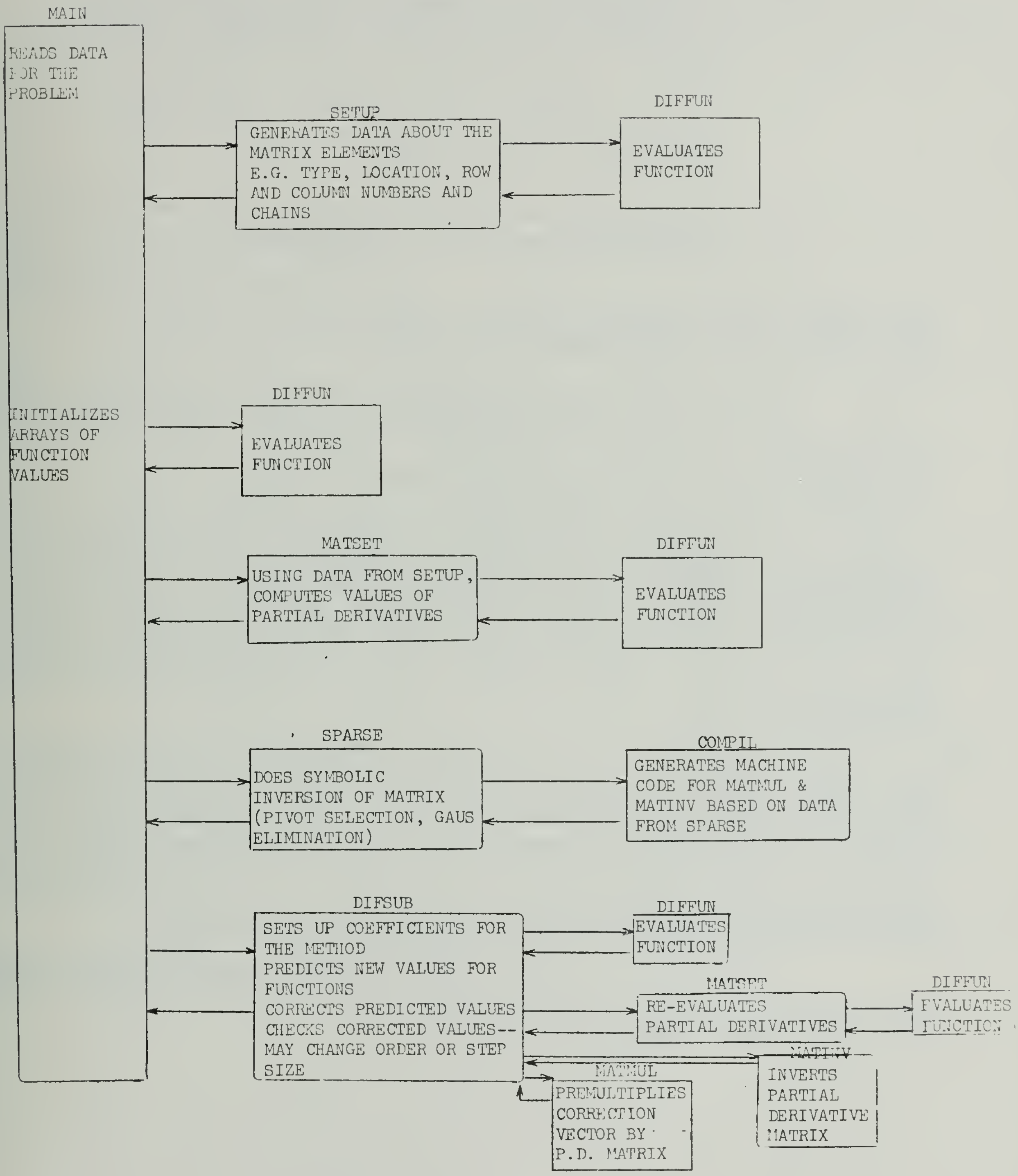

MAIN
READS DATA FOR THE PROBLEM
INITIALIZES ARRAYS OF FUNCTION VALUES
SETUP
GENERATES DATA ABOUT THE MATRIX ELEMENTS E.G. TYPE, LOCATION, ROW AND COLUMN NUMBERS AND CHAINS
DIFFUN
EVALUATES FUNCTION
DIFFUN
EVALUATES FUNCTION
MATSET
USING DATA FROM SETUP, COMPUTES VALUES OF PARTIAL DERIVATIVES
DIFFUN
EVALUATES FUNCTION
SPARSE
DOES SYMBOLIC INVERSION OF MATRIX (PIVOT SELECTION, GAUS ELIMINATION)
COMPIL
GENERATES MACHINE CODE FOR MATMUL & MATINV BASED ON DATA FROM SPARSE
DIFSUB
SETS UP COEFFICIENTS FOR THE METHOD
PREDICTS NEW VALUES FOR FUNCTIONS
CORRECTS PREDICTED VALUES
CHECKS CORRECTED VALUES-- MAY CHANGE ORDER OR STEP SIZE
DIFFUN
EVALUATES FUNCTION
MATSET
RE-EVALUATES PARTIAL DERIVATIVES
DIFFUN
EVALUATES FUNCTION
MATINV
INVERTS PARTIAL DERIVATIVE MATRIX
MATMUL
PREMULTIPLIES CORRECTION VECTOR BY P.D. MATRIX

Figure 2

1. Minimum Row--First Column: The row with the smallest number of non-zero elements is used as the pivot row. The first elgible element in this row is the pivot.

2. Minimum Row--Minimum Column: The row with minimum nonzero element count is pivot row. The element in that row with least nonzero element in its column is the pivot.

3. Minimum Product: The element with the minimum product (elements in row x elements in column) is the pivot.

4. Minimum Column--Minimum Row: Same as method 2 on the transpose of the matrix.

All these methods are dynamic in that at each pivot selection step, the choice depends only on that part of the updated matrix which is in unpivoted rows and columns.

Each method was tested with several examples. However, examples considered were comparatively smaller than those which are expected to occur in practice. Still many striking results and improvements in sparse were achieved.

Pivot selection method 4 was revealed to be the best algorithm of those tested. The word "best" is used here in a restricted sense and means more efficient in a larger number of examples. Method 2 was found to be equivalent to method 4 in efficiency. Method 3, minimum product, was the second best choice as an algorithm, but it must be noted that in two or three examples minimum product proved to be far better than the others.

The strategy of minimum row-first column proved to have very little value as far as efficient pivot selection is concerned. It was, however, invaluable in exposing bugs in sparse. Since this method led to the insertion of many new elements consecutively during the Gaus elimination, errors were detected that would have gone unnoticed in better algorithms which keep the number of new elements to a minimum.

Currently a larger example (50 x 50) is being tested with method 4 and it is felt that this example should be tested with the other algorithms as well.

3.1.2 Steady State Packages (B. van Melle)

The present package is a modification of the numerical integration package. While it works, we are not happy that it is as efficient as possible.

Tests were conducted with the numerical package in an effort to determine a criterion for selecting certain variables in DIFSUB in such a way as to bring about the swiftest convergence in the steady state problem. In particular, the relation between

$$D = \sum_{J=1}^{N} |DY(J)|$$

(where DY(J) is the amount by which the J-th equation is not satisfied) and H (the step size) was examined, as well as the order of the method used.

The tests were conducted by running the problem up to a given order, and then varying the value of H. At each step, several values of H were tried, and the one which produced the greatest reduction in D was selected. This process was continued until D was sufficiently small.

The results for the various examples are given in the following tables. For a given order, the best step size and the accompanying reduction in D are listed for several values of D. The results for smaller values of D are identical to those for the last entry, since the method stabilized for small D. Best results were generally achieved with the low orders, since the method often failed to converge for the higher orders with reasonably large values of H.

The results indicate that the second-order method is good for D larger than .001, with a step size in the vicinity of 1.6, except for rather large D, which require a smaller step size. The linear method with step size of 1.0 is by far the best for small D. The transition from the second order to the first order method is indicated when D is somewhere in the range .01 to .001.

Further experiments are to be made to try to determine a good strategy. In the meantime work has continued on the use of the present package.

3.1.3 Steady State Package Testing (W. Chung, B. van Melle)

The following things have been done as an effort to combine and test the several routines which would constitute a numerical package as a whole.

1. Change of EPS.

2. Reconstruction of Main Program.

3. Test of CMPILE.

4. Test of SETUP.

1. Change of EPS

The error test criterion formula was changed from EPS = 10**(2.3*D - 4.3) to EPS = EXP(2.3*D - 4.3) which was shown by experiments to result in the faster convergence in most cases of the steady state problem.

The comparison of two EPS expressions is given in the table below for several examples (EX1 to EX4).

EPS=	10**(2.3*D-4.3)			EXP(2.3*D-4.3)		
	NS	NFNS	NW	NS	NFNS	NW
EX1	28	96	19	28	96	19
EX2	52	176	26	28	96	19
EX3	43	168	30	35	132	27
EX4	86	490	76	62	363	59

COMPARISON OF TWO EPS'S

The bound for EPS when D is greater than 1.0 was set to 0.1 since EPS = .05 gave slow convergence or no convergence in the new test problem.*

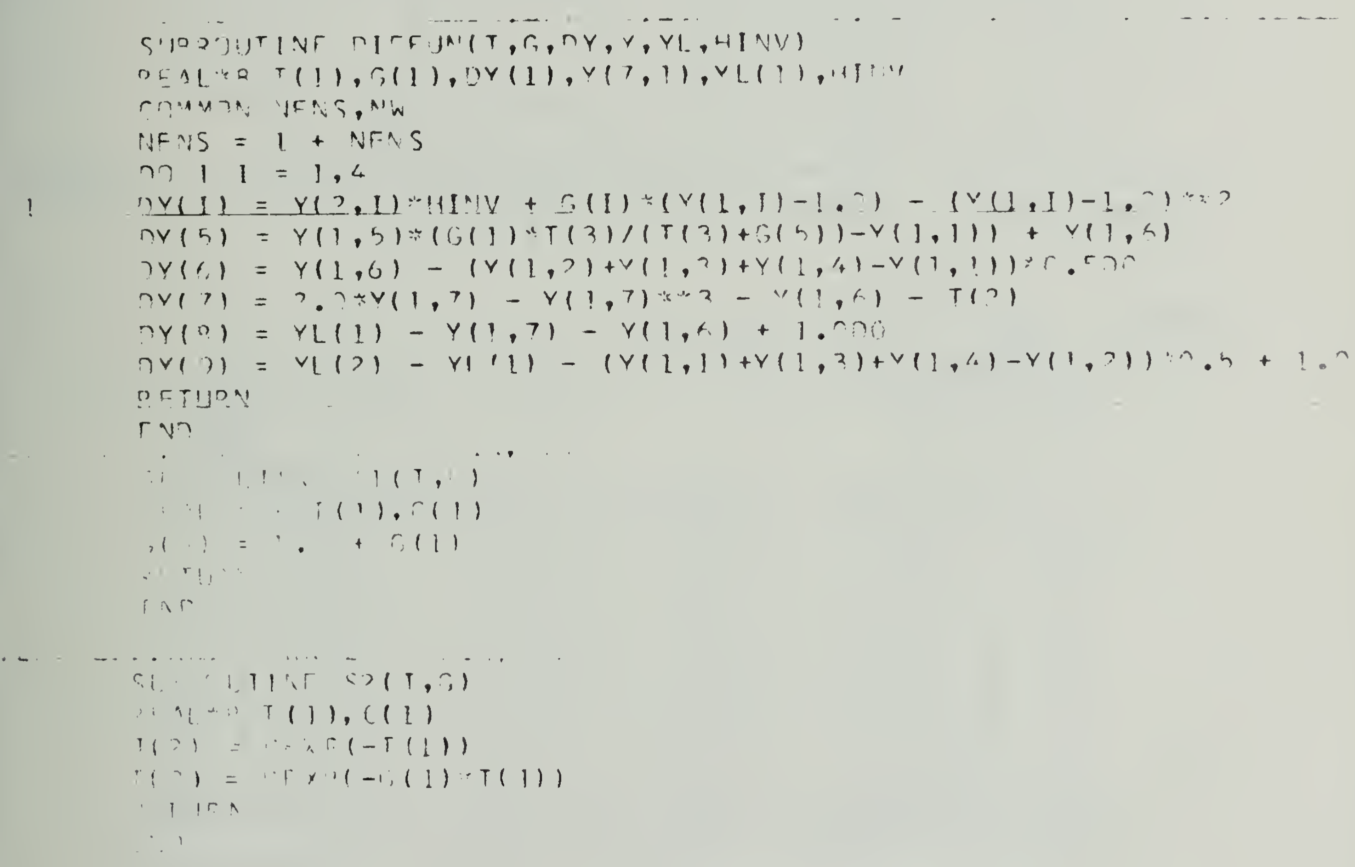
However, this EPS value remains a problem which needs more consideration.

2. Reconstruction of Main Program

The steady state solution of EX6 was obtained after 130 steps of DIFSUB call by using the old main and subroutines.** This was tried to check the subroutine CMPILE which was run together with other new subroutines.***

The first half of the main program was rewritten to match the new subroutine set and have the calling sequence in proper order.

*EXAMPLE 6



```
          SUBROUTINE DIFFUN(T,G,DY,Y,YL,HINV)
          REAL*8 T(1),G(1),DY(1),Y(7,1),YL(1),HINV
          COMMON NFNS,NW
          NFNS = 1 + NFNS
          DO 1 I = 1,4
    1     DY(I) = Y(2,I)*HINV + G(I)*(Y(1,I)-1.0) - (Y(1,I)-1.0)**2
          DY(5) = Y(1,5)*(G(1)*T(3)/(T(3)+G(5))-Y(1,1)) + Y(1,6)
          DY(6) = Y(1,6) - (Y(1,2)+Y(1,3)+Y(1,4)-Y(1,1))*0.5D0
          DY(7) = 2.0*Y(1,7) - Y(1,7)**3 - Y(1,6) - T(2)
          DY(8) = YL(1) - Y(1,7) - Y(1,6) + 1.0D0
          DY(9) = YL(2) - Y(1,1) - (Y(1,1)+Y(1,3)+Y(1,4)-Y(1,2))*0.5 + 1.0
          RETURN
          END

          SUBROUTINE S1(T,G)
          REAL*8 T(1),G(1)
          G(1) = [illegible] + G(1)
          RETURN
          END

          SUBROUTINE S2(T,G)
          REAL*8 T(1),G(1)
          T(2) = DEXP(-T(1))
          T(3) = DEXP(-G(1)*T(1))
          RETURN
          END
```

** MATSET, DIFSUB, MATINV, MATMUL

*** DIFSUB, SETUP, SPARSE, COMPIL

EXAMPLE 1

$$F(x) = \begin{cases} \sin(x_1 x_2) - x_2/\pi - x_1 \\ (1 - \frac{1}{4}\pi)\ (\exp(2x_1) - e) + ex_2/\pi - 2ex_1 \end{cases}$$

	BEST H FOR THE GIVEN ORDER AND THE ACCOMPANYING REDUCTION IN D						BEST ORDER FOR GIVEN VALUE OF D	
	1st ORDER		2nd ORDER		3rd ORDER			
D	H*	RED	H	RED	H	RED	ORDER	H
1.0	1.1	.51	0.9	.37	0.9	.46	2	0.9
.3	2.0	.35	1.4	.06	1.3	.42	2	1.4
.1	4.0	.21	1.6	.04	1.3	.40	2	1.6
.03	4.0	.20	1.6	.03	1.3	.39	2	1.6
.01	4.0	.20	1.6	.03	1.1	.17	2	1.6
.003	1.0	.0007	1.8	.12	1.3	.04	1	1.0
.001	1.0	.0006	0.8	.60	1.3	.04	1	1.0
.0003	1.0	.0005	0.8	.60	1.3	.04	1	1.0

* H--the step size. Values tried for H ranged from .2 to 4.0
RED--reduction in D, which is equal to (new value of D)/(odd D)

EXAMPLE 2

$$F(x) = \begin{cases} x_1^2 - x_2 + 1 \\ x_1 - \cos(\pi x_2/2) \end{cases}$$

	1		2		3		4		BEST ORDER	
D	H	RED	H	RED	H	RED	H	RED		H
1.0	0.9	.46	1.6	.16	1.6	.24	0.5	.64	2	1.6
.3	1.1	.40	1.6	.15	1.8	.23	1.6	.11	4	1.6
.1	1.4	.36	1.6	.10	1.8	.25	1.8	.17	2	1.6
.03	2.0	.30	1.8	.06	1.6	.31	1.8	.14	2	1.8
.01	4.0	.17	1.6	.03	1.4	.35	1.8	.08	2	1.6
.003	4.0	.24	1.6	.02	0.9	.33	1.8	.03	2	1.6
.001	1.0	.002	1.6	.02	1.3	.05	1.8	.02	1	1.0
.0003	1.0	.0007	0.8	.60	1.3	.05	1.3	.39	1	1.0

EXAMPLE 3

$$F(x) = \begin{cases} 4 + x_1 + x_2 - x_1^2 + 2x_1x_2 + 3x_2^2 \\ 1 + 2x_1 - 3x_2 + x_1^2 + x_1x_2 - 2x_2^2 \end{cases}$$

D	1		2		3		4		BEST ORDER	
	H	RED	H	RED	H	RED	H	RED		H
2.0	0.9	.62	0.7	.52	0.7	.50	0.4	.77	3	0.7
1.0	1.8	.42	1.3	.12	0.6	.65	0.6	.65	2	1.3
.3	4.0	.21	1.4	.04	1.3	.43	1.3	.32	2	1.4
.1	4.0	.15	1.6	.05	1.3	.40	1.6	.08	2	1.6
.03	4.0	.20	1.6	.04	0.9	.33	1.6	.07	2	1.6
.01	1.0	.007	1.6	.03	1.1	.17	1.8	.05	1	1.0
.003	1.0	.007	2.0	.22	1.3	.04	2.0	.19	1	1.0
.001	1.0	.006	0.8	.60	1.3	.05	1.3	.40	1	1.0

EXAMPLE 4

$$F(x) = \begin{cases} x_1^2 + x_2^2 + x_3^2 - 5 \\ x_1 + x_2 - 1 \\ x_1 + x_3 - 3 \end{cases}$$

D	1		2		3		4		BEST ORDER	
	H	RED	H	RED	H	RED	H	RED		H
2.0	1.1	.41	1.3	.22	1.3	.30	1.6	.15	4	1.6
1.0	1.3	.39	1.8	.09	1.8	.27	1.4	.25	2	1.8
.3	1.8	.32	1.8	.04	1.6	.31	1.8	.14	2	1.8
.1	3.0	.22	1.6	.03	1.4	.35	1.8	.09	2	1.6
.03	4.0	.24	1.6	.01	1.4	.37	1.8	.03	2	1.6
.01	4.0	.21	1.6	.02	1.0	.26	1.8	.01	4	1.8
.003	1.0	.004	1.6	.02	1.3	.05	1.8	.02	1	1.0
.001	1.0	.003	0.8	.60	1.3	.05	1.3	.40	1	1.0
.0003	1.0	.003	0.8	.60	1.3	.05	1.3	.40	1	1.0

3. Test of CMPILE

To check whether CMPILE generates the correct codes for DIFFUN, S1 and S2, two programs were run, one using CMPILE and the other using hand coded subroutines DIFFUN, S1 and S2 without CMPILE. Through this some trivial errors in CMPILE were found and corrected, then finally two programs produced the same results. The fact that results of these programs were identical with that of previous program (old version) means that CMPILE, SETUP, and SPARSE are working nicely.

4. Test of SETUP

The SETUP written by K. Ratliff has been used through the above tests. Finally, another version of SETUP by A. Whaley was available and tested to see if it is working properly. They gave the same solution, however, after different numbers of function calls. (154 steps for the old SETUP and 168 for the new one.) The difference is caused by the different techniques used in SETUP to determine when a partial derivative is a constant. After some changes, the new set up produces a simpler matrix for the Jacobian.

3.2 Non-Numerical Packages (A. Whaley)

3.2.1 Compiler

The compiler, which has been working, was improved to generate more efficient object code. Both general registers and floating point registers are obtained for use from subroutines which allocate the register which is unused for the longest period of time. These changes will make the produced object code about 5/6 as long as that previously obtained. The former version only used one general register and one floating point register. The routine which handles the operators +, -, *, / is careful to take advantage of the possibility that some of the operands are already in registers. For + and *, if the second operand is in a register and the first is not, the sense of the operation is reversed (i.e., A+B to B+A) to allow the operation to be performed immediately. Intermediate results are never stored unless their register is required for some other operation.

3.2.2 Filing System

The filing system is fairly well debugged at this point and seems to be working satisfactorily. A command to delete records was added. As the previous write-up was not very clear, and since that time most of the rules have changed, a hopefully complete write-up of the filing system follows:

```
          XINIT  <XLIST>
```

Here XLIST should be replaced with the name of an XLIST parameter area. In this special case, the XLIST is not used as a parameter list but as a scratch work area. This command allows a user to access the files residing in an OS dataset. The dataset is defined by a DD card with the name DISK. An example defining a temporary data set is as follows:

```
          //DISK  DD  UNIT=DISK,SPACE=(TRK,1Ø)
```

Additional parameters:

```
          XINIT  <XLIST> INIT,FORMAT
```

FORMAT will format the dataset for use, placing eight blocks on each track of 793 bytes each. INIT and FORMAT initialize the disk to have no files existing and all space free and available. Formatting is only required before the first time that a dataset is used.

After XINIT is complete, the user will discover in R1 the address of something called the XDSCB. This contains all the pointers, temp storage, etc., for the re-entrant filing system program. This address must be in every XLIST to be used. It may be placed there from register one as follows:

```
          XLINK <XLIST>XDSCB=(R1)
```

It may also be obtained from another XLIST that already has this address:

```
          XLINK  <XLIST> LIST2=XLIST

          OPEN   <XLIST> NAME=
                         BUF=
                         NREC=
                         ABEND=
                         BUFLEN=
                         TEST=
```

This command opens a file. Afterwards, until it is closed, it may be accessed by referring to the same <XLIST>. The NAME parameter is for specifying the file name. If the file did not previously exist, it will be created. Abnormal conditions for this macro are SEQ which is signaled if the file appears to already be open.

BUF is for specifying a buffer address. This parameter is not used for the open,but any parameter coded on any X macro will be there for execution of other X macros unless overwritten. (This statement applies only to BUF, NREC, BUFLEN, and NAME.) BUFLEN is the length of the buffer. NREC is the number of records to be read or written during XREAD or XWRITE. Severe abnormal conditions that are not tested will result in a branch to a location called ABEND (e.g. XYZ) the parameter ABEND=XYZ may be coded.

```
XREAD    <XLIST>,NREC=
                 BUF=
                 BUFLEN=
                 ABEND=
                 TEST=
```

The first eight bytes of the buffer provide the line number of the first record to be read. Remaining records are read in a sequential fashion. Each record is placed one after the other and begins with the appropriate eight character line number and a two byte length of the remaining data. The source record format must be provided by the user when doing a write. Abnormal conditions which may occur are SHORT for a too-short buffer, RNF for a record not found, EOF for end of file, PARM for parameter error, and SEQ for sequence error (i.e., file not yet opened). For a more elaborate discussion of how to code the TEST parameter, see the previous write-up of the filing system. If abnormal conditions occur, <XLIST> + 34 contains the number of records successfully transferred.

```
XWRITE   <XLIST>,NREC=
                 BUF=
                 BUFLEN=
                 ABEND=
                 TEST=
```

Write the number of records specified, each having a record name and remaining data length. Abnormal conditions that may be checked are: PARM, SEQ, and RF (record already exists). For explanations, see XREAD. Abnormal conditions cause the operation to be terminated. Number of records successfully transferred is in <XLIST> + 34 (halfword).

```
XDELETE  <XLIST>,NREC=
                 BUF=
                 BUFLEN=
                 ABEND=
                 TEST=
```

Causes one record, whose name is in the first eight bytes of the buffer, to be deleted from the file. Abnormal conditions are RNF, record not found.

```
XDEST   <XLIST>,NAME=
                NREC=
                BUF=
                BUFLEN=
                ABEND=
                TEST=
```

The <XLIST> named must not be associated with an open file. The file name by NAME= (here or previously) will be destroyed. Abnormal conditions are FNF (file not found) and SEQ (<XLIST> is being used with an open file).

```
<XCLOSE>  <XLIST>,NREC=
                  BUF=
                  BUFLEN=
                  ABEND=
                  TEST=
```

Closes open file associated with this <XLIST>. Abnormal conditions are SEQ if the file is not open. Also causes any blocks which have not yet been actually moved from core to disk to be so moved.

```
XTERM   <XLIST>
```

Undoes an XINIT: fremains control blocks used by' the filing system, closes the OS dataset, etc.

3.2.3 Item Analysis (J. Koch)

The item analysis routine takes the data structures generated in the PDP-8 and produces a block which is used as input for Global Analysis. In Item Analysis such errors as illegal syntax in equations or declarations are detected and messages are sent to the user. Item was modified so it now eliminates from the node/connection table used by Global Analysis a terminal connected to itself, duplications of termianl connections and loops of connections. Item was also modified to be consistent with what is expected as input for Global Analysis. It uses the graphics filing system to save copies of the structures sent up from the PDP-8. These are retrieved from the filing system when the command 'ITEMIZE name' is given. Thus, once an element definition has been sent from the PDP-8, it can be used without having to send it up with each network that contains it. The filing system is also used to save the output from Item Analysis in ITEMLIB where it is then called in by Global Analysis. Primitive elements and networks have now been created in the PDP-8, sent up to Item Analysis and passed on through Global Analysis successfully. Item Analysis uses a routine called PARSE to parse the equations. This routine has been modified to handle larger equations and will be able to handle floating point constants in the next quarter.

3.2.4 Global Analysis I and II (S. Wilkins)

A summary of the goals of the Global Analysis in its final form:

Figure 3 is a chart showing the position of Global Analysis I and II in the calling sequence of the Simulation and Modeling System. Global I can be activated by Item Analysis after storing data in the filing system or by the user whenever he wishes to analyze a previously defined network. Once Global I is activated no modifications to the user's data are allowed during the analysis. The only exception is that the user may supply different sets of values to variables global to the whole network, since these do not affect the structure of the network.

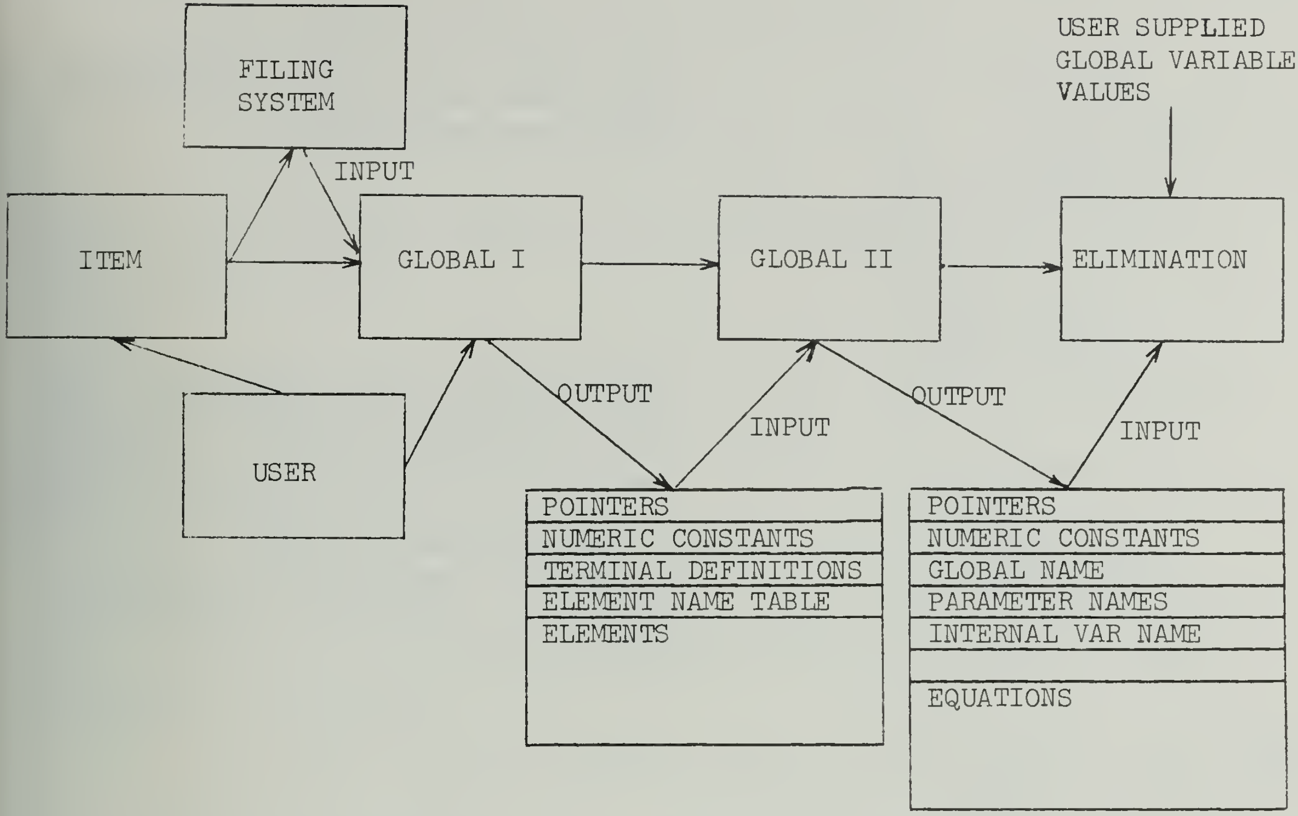


Figure 3

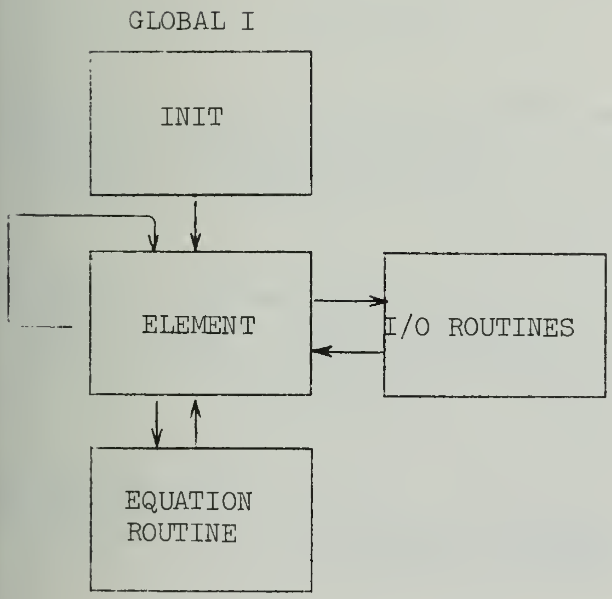


Figure 4

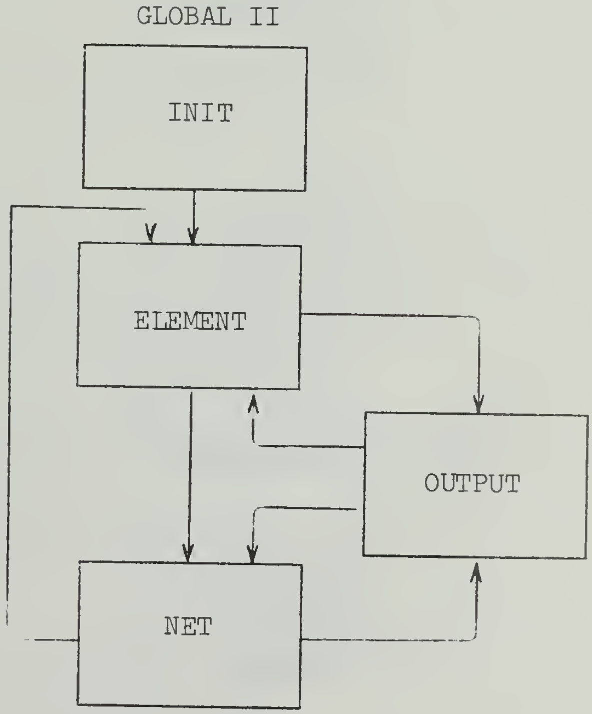
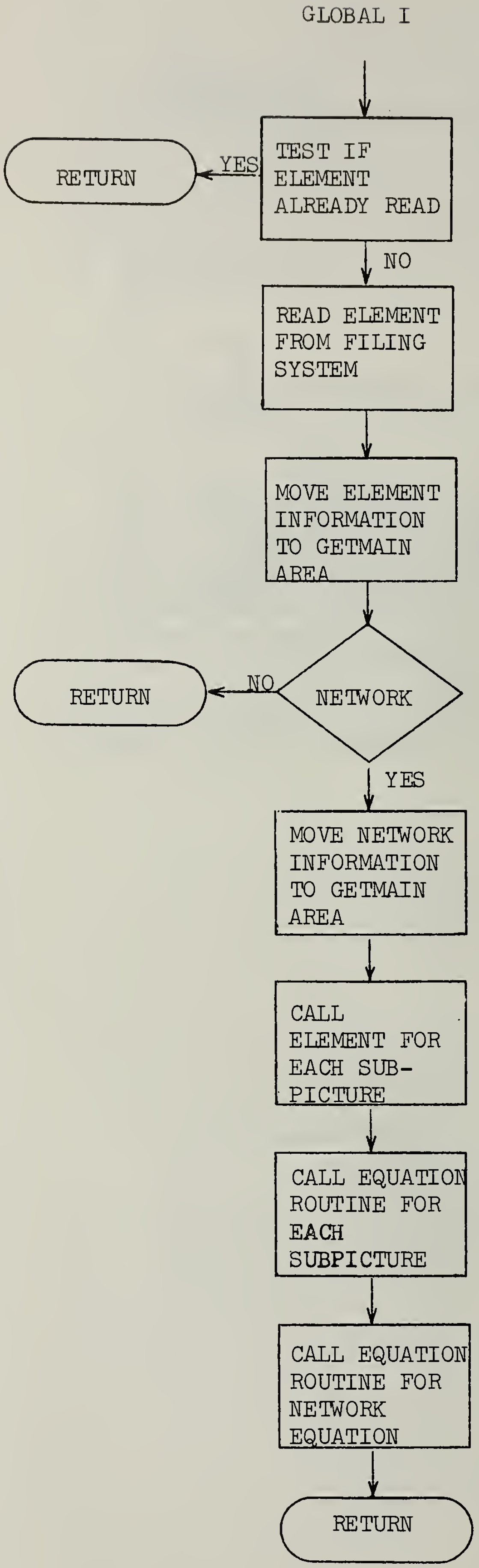


Figure 6


GLOBAL I
TEST IF ELEMENT ALREADY READ
YES
RETURN
NO
READ ELEMENT FROM FILING SYSTEM
MOVE ELEMENT INFORMATION TO GETMAIN AREA
NETWORK
NO
RETURN
YES
MOVE NETWORK INFORMATION TO GETMAIN AREA
CALL ELEMENT FOR EACH SUB-PICTURE
CALL EQUATION ROUTINE FOR EACH SUBPICTURE
CALL EQUATION ROUTINE FOR NETWORK EQUATION
RETURN

(Every network is stored as an element) Each element that occurs in the network need be analyzed only once.

The element is added to the Element Name Table and copied into the Element Area with certain modifications. All symbolic references to terminal types are replaced by internally assigned names. All numeric constants are placed in the constant table and are referenced by their position in the table. The element is assigned a numeric name.

This includes descriptions of subpictures and the construction of the network.

Each subpicture or element in the network must be analyzed and assigned an internal numeric name and then terminal types on all the connections must be checked for consistency.

The equation routine searches all equations on each subpicture for references to terminal variables and assigns to them internally defined names.

The equation routine is also used for all equations associated with the network.

Figure 5

Figure 4 is a flowchart of the routines comprising Global I. The function of 'Init' is to

1. Initialize a getmain area* in which various tables are built for use by Global II.

2. Setup the initial call to 'Element' with the name of the network to be analyzed.

'Element' is a recursive routine that reads element and terminal definitions from the filing system and builds tables in the getmain area. 'Element' operates on the network in a top-down manner.

Figure 6 is a flow chart of the routines in Global II. The function of Global II is to produce a set of equations equivalent to the network being analyzed. 'Init' initializes the getmain area* in which tables and equations for use by 'Elimination' are constructed, and sets up the call to 'Element.'

'Element' analyzes the network for equations in a topdown manner just as in Global I. Both 'Element' and 'Net' are recursive routines. 'Element' calles 'Net' when an element is a network and 'Net' calls 'Element' to analyze each subpicture element in the network. A detailed flow chart of Global II is given in File #824.

The main tasks of Element are

1. To assign internal names to global and internal variables and parameters.

2. To assign parameter values to the proper parameters.

3. To call 'Output' to modify and output the element and parameter equations.

The main tasks of 'Net' are

1. To assign unique terminal variable names at each node in the network using Kirchoff's current law to determine the assignment of I-type variables and Kirchoff's voltage law to determine the assignment of E-type variables.

2. To generate the equations necessary to show the relation between I-type variables at each node.

3. To call 'Output' to modify and output any network equations.

* The data structures constructed by Global I and II are described in detail in previous reports.

4. Call 'Element' for each subpicture in the network supplying the correct E- and I-variable assignments for the subpicture terminals and any parameter values.

3.2.5 Elimination (J. Frazao)

A package now called WEED has been written. WEED is an algorithm for symbolically simplfiying systems of equations in treestructured form and classifying variables for the later numerical analysis stage. Through a consistent method of elimination, WEED produces a condensed system of equations and an Equivalence Table (EQV) for variables, which allow for a more efficient compilation and numerical integration than would otherwise be possible.

A number of passes are made through the system of equations, each time advantage being taken of simplifications made in previous passes. As each equation is processed, all expressions which are only functions of constants are evaluated and replaced by the result. Expressions of the form

E+O , O+E, E*O, O*/E, E*/1, 1*E, E↑O
E↑1, 1↑E and d/dt(S1 or S2 expression)

are simplified as they are detected, and unary minuses are eliminated, combined with other unary minuses or carried up the tree wherever possible.

Equations which become trivial relations, such as

variable = constant

or

variable = ± variable

are eliminated by setting appropriate pointers in the EQV, and the remaining equations are placed into three sets:

1. S1-equations, of the form:

   variable = fn (global vars & consts only)

2. S2-equations, of the form:

   variable = fn (globals, consts and TIME)

3. General equations (those which are neither S1 or S2), of the form:

   expression = 0.

In addition, as variables are encountered in the equations, they are classified into four sets according to their usage:

1. Set S1: variables defined by S1-equations.

2. Set S2: " " " S2-equations.

3. Set L : variables used only in linear algebraic expressions.

4. Set M : all other variables.

By performing these classifications, each set of variables and each set of equations can receive special treatment in the numerical integration phase, thus allowing for a more economical program execution.

The equations are reprocessed until there are no more changes in the EQV (up to a maximum of three passes). All variables are then renumbered sequentially, with all S1-variables first, then S2, L and M variables. The new names are substituted in the equations, and these in turn are compressed and passed on (in their three different sets), along with the EQV, to the numerical integration routines.

3.3 Graphical Remote Access Support System (GRASS)

3.3.1 Disk Monitor System (C. Hyde)

The new disk monitor on the new disk interface became operational this quarter and has performed very well. (c.f. section 3.4.3).

3.3.2 Display Terminals (R. Haskin)

ACID, the display terminal controller, has undergone several important modifications this quarter.

The first of these was changing IOT assignments to accommodate the terminal multiplexor. Sections of code which previously simulated this multiplexor were removed and replaced by the actual device IOT's. After the multiplexor hardware was tested, ACID was tried for the first time with two terminals, and worked perfectly.

Another modification was made to support the 8/I teletype as an operator's console. The function of the console is to put the terminals on and off line so inactive terminals will not interfere with the system.

Strategy changes were made in the methods used to display pictures from the PDP-8. The most important of these was changing the program so that the keyboard is enabled for input between segments of display files rather than only after the entire display file has been received. This enables all pictures to be sent as segments, which eliminated the time spent waiting on completion of the last picture regeneration. The time it now takes to regenerate a reasonably complex picture (standard frame, about 20 nodes and 150 lines) is 1 to 1.5 seconds, which is considerably faster than previous expectations.

3.3.3 Information Retrieval Package (M. Michel)

The local IR package has been in use all quarter as part of GLASP and has run reliably. Copies of the program and documentation have been given to a research group at Griffiss Air Force Base, Rome, New York, to help them in development of their own graphics facility.

3.3.4 Disk Interface (C. Hyde)

The new disk interface became operational this quarter and has been put into full production use, both for the disk monitor system and for GLASP (c.f. section 3.4.3).

3.3.5 Monitors (M. Michel)

Multi-terminal support by both GLASP and GRASP is now operational (!).

GRASP (360 Remote Monitor)

The top-most monitor level, G8OPERAT, has been in use all quarter. All commands work as specified, and no abnormal terminations have occurred.

The second-level, multi-terminal monitor, SPACT, has also been in use all quarter. All commands work as specified, and two new ones have been added. '!DUMP' takes a snap of the current task attached to the requesting terminal; '!DSPACT' takes a snap of SPACT itself and the filing system module. A few bugs turned up at the beginning of the quarter. These were corrected and subsequently, SPACT has successfully handled a number of different user tasks on different terminals simultaneously. These include the simulation and modeling system (specifically the module ITEM), and several utility packages (REPLYRE, COMUNE, LSD, etc., c.f. section 3.3.6). The user-level macros and routines for 2701 communication to the terminals (S8READ, S8WRITE, S8SETUP, etc.) have been in use successfully all quarter.

2701 Data Link

Usage increased again this quarter as debugging of user-written packages running under the monitor increased. Operation has been very good, although dense transmission of very long records has not occurred. One intermittent problem has begun lately (a premature time-out status at the 360 end) but has not happened often enough (yet) to be pin-pointed.

GLASP (PDP-8 Local Monitor)

GLASP has continued to be reliable this quarter, and no changes have been made. Multi-terminal operation was begun and worked perfectly the first time tried.

GLASP Service Routines

All services have been in heavy use this quarter except for Inktronic printing. No bugs have been found.

Additions to handle creation, integration, and display of subpictures, subpicture instances, and mnemonics were made, debugged, and put into production use during the quarter.

GLASP Program Segments

GENDRW was expanded with additional functions including basic subpicture handling, remote picture transmission, parameter assignment, etc. These new functions were provided mainly by making calls to GUT1 and GUT2 for access to "common" routines.

COMUNE was rewritten, expanded, and renamed REACT. It is now a very flexible tool for easy communication of commands and data structures to and from the remote CPU (c.f. section 3.3.6 on COMUNE and LSD). In general, data structures (pictures) and mnemonic definitions are transmitted to the 360 for archival storage. On request, these items can be variously

1. Fetched back to the PDP-8 and automatically stored on the local filing system.

2. Fetched back but not stored.

3. Displayed on the user's console.

Note that 1 and 2 cause the fetched item to replace the user's current working data structure in the PDP-8. After a fetch, the user can enter GENDRW and modify the item. But 3 only causes a display of the item, the current PDP-8 data structure remains unaltered. Any of 1, 2, or 3 are used in conjunction with LSD for examining, storing, and retrieving items from the remote permanent library. Mode 3 is mainly for use in conjunction with tasks like COMUNE for allowing access to the terminals in a "transparent" manner (e.g. the system appears as a terminal hooked to the remote CPU without the PDP-8 and PDP-8/I.

GLASP Status

Phase 0.5 has been reached: a skeletal system for stand-alone or remote interactive graphics support has been created. At this point, there is a great need for producing complete documentation for the elements of the system now implemented. In addition, although all aspects of this very initial version are operational, full testing and complete debugging has not yet taken place. While a rapid development and implementation of features occurred last quarter and this quarter, the coming quarter will be devoted to clean-up, consolidation, and debugging. Visibly moving above phase 0.5 will probably not occur until some time into the third quarter of this year.

Simulation and Modeling (SAM) System Support

GRASS is now being used actively for the development of the SAM System. Primitives and various networks using these primitives are created locally and transmitted directly to the 360 filing system (by LSD or ITEM) or are saved locally on magnetic tape for later transmission. ITEM is now running as an interactive user task in the 360, using the 2701 transmission facilities of the monitors for I/O to the terminals, using the 360 filing system for network (picture) and analysis storage, and calling the next level of SAM (Global).

A batch driver for ITEM was written and put in use this quarter, so that some types of debugging can be done from the system batch job stream.

A back-up tape of the 360 filing system is being maintained in case of system failure.

3.3.6 Remote Data Structure Utilities (R. Haskin, J. Nickolls)

Data Structure Communications Package

COMUNE, the communications and data structure handling program, was debugged further this quarter, and several changes have been made, including the ability to send data structures to the filing system and to plot them. Also, several calling sequences to current routines have been changed.

Changes to Current Routines

INIT: INIT has been removed and all routines in the COMUNE package are self-initializing.

TEXT: CALL TEXT(1X,1Y,'TEXT',N)

Text creates an entry in the text block consisting of the text string 'TEXT.' The first character starts at point (1X, 1Y) (in increments from the lower left hand corner of the picture). N is the number of characters in the string. If IX = IY = 0 and N is positive, the new text line appears immediately below the previous one. If IX = IY = 0 and N is negative, the new text line is concatenated with the previous one.

SEND: CALL SEND(IBLK,ICODE)

Send Block IBLK (IBLK = -1 means send all initialized blocks) to the PDP-8. ICODE is an INTEGER*4 array dimensioned 5 whose elements are as follows:

ICODE(1)--1 = reinitialize blocks when done sending
0 = retain blocks

ICODE(2)--1 = replace block(s) sent in PDP-8 data structure
0 = retain current PDP-8 data structure

ICODE(3)--1 = do not display the block(s)
0 = display them

ICODE(4)--1 = erase screen before displaying block(s)
0 = add blocks to current screen image

ICODE(5)--1 = put up frame before displaying block(s)
0 = do not put up frame

```
MASK:      CALL MASK(IMASK)

          IMASK is an integer array, dimensioned 11, whose
elements are as follows:

          IMASK(1)--1 = do not echo a user-typed text
                        line on the display
                    0 = echo each user-typed text line

          IMASK(2) through IMASK(10)--for screen segments
                        1 through 9
                    1 = accept joystick input
                    0 = ignore joystick input

          IMASK(11)-1 = accept a user-typed text line
                    0 = ignore a user-typed text line
```

Hard Copy Output

Several modifications to the hard copy output routines have been made. The routines now take the data for the pictures to be plotted from the filing system, allowing a HASP job to be run to do the plotting off line. Programs in the graphics system can request hard copy output by first storing the data structure of the picture in the filing system, and then putting the name of the file containing the picture into file PLOTLIB in the filing system. COMUNE and LSD now have facilities in them to request hard copy output.

Graphics Library Maintenance Program

A program to provide facilities for storage and retrieval of data structures in the 360 filing system has been written and debugged. It allows pictures to be sent from the PDP-8 and saved, or fetched from the filing system and sent to the PDP-8. It also has provision for allowing pictures to be plotted by the hard copy output routines. This program, the Library Service Discographer (LSD), has the following facilities:

Saving of Items

```
SA[VE]  <picture,name>
        <mnemonicname>.<mnemonic id #>
```

Save allows a picture or menmonic to be sent from the PDP-8 and saved in the filing system. The procedure for doing this is to first type in the save command, and then, after LSD replies 'SAVE READY', type in the PDP-8 command '##<picturename>' which will cause the picture to be sent to the 360. The 'mnemonic id number' is a one digit number from 0 to 7, which allows several versions of a mnemonic (such as horizontal and vertical resistors) to be stored under the same mnemonic name. The new picture or mnemonic replaces any previous occurrence in the filing system.

Fetching of Items

```
FE[TCH]  <picturename>
         <mnemonicname>.<mnemonic id>
```

Fetch retrieves the specified picture or mnemonic and sends it to the PDP-8, where it replaces the current picture in the data structure. LSD then replies with a message to indicate whether or not the picture was found.

Display of Items

```
DI[SPLAY]  <picturename>
           <mnemonicname>.<mnemonic id>
```

The action of DISPLAY is similar to that of FETCH except that the picture will be displayed at the terminal, and the current picture in the PDP-8 data structure will not be replaced.

Deletion of Files

```
DE[LETE]  <picturename>
          <mnemonicname>.<mnemonic id>
```

Deletes the picture or mnemonic instance in the library.

Plotting Files

```
PL[OT]  <picturename>
        <mnemonicname>.<mnemonic id>
```

PLOT puts the name of the item to be plotted into a file 'PLOTLIB' in the filing system. When the hard copy output program is run later from a HASP job, the item will be plotted.

Copying Files

```
CO[PY]  <picturename 1>
        <mnemonicname 1>.<mnemonic id 1>

                 <picturename 2>
                 <mnemonicname 2>.<mnemonic id 2>
```

The first item is copied into the second file, anything previously in the second file being overwritten.

Auto, Dauto

```
AU[TO]
DA[UTO]
```

The commands turn on and off the 'automatic save' feature in the PDP-8. When AUTO is in effect and a picture is sent to the PDP-8 via a FETCH or DISPLAY command, the picture is saved in the PDP-8's local IR. DAUTO prevents this from occurring.

Logging Out

```
GO[ODBYE]
```

Causes LSD to close all files and terminate. A message is sent to the user to inform him of the completion of this termination.

3.3.7 Other Utility Software

External GLASP Utilities

This quarter, several programs were written to facilitate loading and storing the GLASP system and its local library.

GLOAD: The GLASP system can now be easily loaded from a GLOAD tape, which has the program segments and the system programs, each with a program number. The load program, GLOAD, loads the 32 program segments from tape onto disk, then loads the core resident portions of GLASP. As an option, the local disk library can either be freshly initiated or restored as of the last GEXIT (see below) from a GLASP Library Tape. An additional option allows loading the PDP-8/I with ACID (the display controller) via the PDP-8/PDP-8/I data channel.

GEXIT: The local disk library may be saved on the GLASP Library Tape by bringing up the dectape system and calling GEXIT which saves the data and pointers for a subsequent GLOAD.

WLOD: The GLOAD tape may be updated with new assemblies from the PDP-8 disk monitor system by calling WLOD, specifying a system disk as input, and the GLASP program segment number as output.

PEEPER: The GLOAD tape may be directly modified by using PEEPER from the dectape system. The user specifies a program segment number, and then he may change any instruction in that program directly on the GLOAD tape. An option allows modification of any specified general tape block.

Computek Test Routines

An alignment and test program for the Computek CRT graphics terminals was written. It runs in the PDP-8/I, and has 17 looping displays for visual alignment, and four static displays for specific voltage measurements. The items checked are: loop time constant, offset, overall gain--horizontal and vertical, overall gain, velocity sensor, slow loop offset, fast loop offset, fast loop DC gain, overall offset, slow loop DC gain, time constant, and scope.

3.4 Hardware

3.4.1 Computek Computer Graphics Terminal (C. Carter, H. Lopeman, R. Miller)

STATUS:

University of Illinois P. O. #215711 (First Complete Unit)

Most of the problems have been remedied and is now considered operational.

University of Illinois P. O. #228151 (Operating Spares)

The 611 has been returned, checked out and considered operational and under warranty.

University of Illinois P. O. #233131 (Second Complete Unit)

The 611 has been repaired and warranty has begun on this unit. The terminal is considered operational.

We recently received from Computek many of the promised materials by Mr. Bob Cvitkovich. They are:

1. One extender card.

2. New service manual.

3. Spin-tight tool.

4. Sylvania IC information (in service manual).

5. Spare parts list (in service manual).

The slash thorugh the zero (Ø) modification has been done in house simply by adding a 10K 1/4W resistor at P4 time to intensify the beam.

SPARES:

Of the 23 cards mentioned in the last quarterly report to be built as replacement cards, 22 of these have been built and checked out. The card to be built is card 'C'.

CONSTRUCTION:

1. Three cables have been wired and two checked out which interfaces the two terminals with the PDP-8/I multiplexor.

2. Rack has been wired for checkout of the 23 above mentioned cards and may be used for further design changes while the two terminals are in use.

3. The multiplexor has been checked out and is operatio

4. The Joystick interface (card 'f') has been built and checked out using a 'Mouse' for the joystick.

5. Curve generation is now being undertaken and card F has been modified, card I has had the drivers added, card J (containing the 'fast loop') is in the process of being built, and card K has a few modifications that need to be added.

3.4.2 PDP-8 System Engineering Log Summary

TTY

1. TTY grinding, very noisy. (Repaired)

DISK

1. Problems with Track 4, Sector 0. (Fixed)
2. Clock track failing on disk. (Pulled disk down and repaired clock track amplifier.)
3. Lost disk logic power supply. (Repaired broken bus.)
4. Disk track 13 not swapping. (Switching to Track 0.)

INKTRONIC

1. Inktronic dead. Pump screen clogged. Ink level in tank too high. (Cleaned and repaired.)
2. Inktronic paper feed dynamic braking control burned out. (Repaired)

PDP-8

1. Switches on PDP-8 register intermittant. (Replaced all known bad ones.)
2. Top DECtape transport has slow response (tape motion) problem is intermittant.

PDP-8/I

1. Several indicators burned out. Some drivers gone. (Replaced bulbs and driver transistors.)

338

1. 338 problem in deflection drive circuits (X) bad board in A19. (Located and replaced.)

COMPUTEK

1. #1 terminal had joystick problems. (Repaired)

3.4.3 Disk (C. Hyde)

The new disk interface became fully operational during the first weeks of the quarter. At that time, all users of the PDP-8 disk system reformatted their swap tapes. One minor error was discovered and eliminated in the disk monitor software.

One hardware error was detected and corrected during this period.

3.4.4 Multiplexor

The multiplexor for the Computek terminals was completed and tested. Several minor construction errors were corrected, and the unit is now operational with up to three device slots currently available. A report which describes the operation of the multiplexor is being written.

3.4.5 Line Buffer

The design of the Computek line buffer is about 90% completed. Construction and testing should begin as soon as a third terminal is available.

3.4.6 PDP-8/I, Stereomatrix Interface (I. Cunningham)

The design of the interface between the PDP-8/I and the stereomatrix display has been completed and all of the cards have been layed out. The back plane wiring has been started and one card is almost complete.

Transmission methods between the display and the PDP-8/I have been finalized. Control signals and picture coordinates are transmitted using SN55107 series line drivers and receivers. These are TTL compatible, fast, and insensitive to external noise. The cursor coordinates are transmitted by coaxial cable and converted to 10 bit words at the PDP-8/I. The coordinates exist internally in the display as analog values and placing the A/D's at the computer saves in cable costs.

The cursor for the stereomatrix is generated by the display logic. The coordinates are stored in a buffer in the PDP-8/I with each display frame and are available to the program at any time. Two interrupt flags are associated with the cursor. The first flag CURSOR indicates that the computer should use the present position for its next operation. The second flag, INCIDENT, results when the cursor approaches "close" to a line on the screen that is to be identified. The logic that generates the display picture pauses when the INCIDENT flag is set. This permits the program to read the display memory address which identifies the line being drawn. Then picture generation can be resumed with no loss of data.

The interface design permits the addition of a programmer defined function box with up to ten buttons plus interrupt.

The hardware should be completed in the next quarter and the development of the software begun.

Form AEC-427
(6/68)
AECM 3201

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO. COO-1469-0182	2. TITLE 1st QUARTERLY REPORT 1971 (January, February, March)

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report

□ b. Conference paper not to be published in a journal:

Title of conference ______

Date of conference ______

Exact location of conference ______

Sponsoring organization ______

□ c. Other (Specify) ______

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.

□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.

□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

C. W. Gear, Professor
and Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois

Signature Charles W Gear	Date March 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.

□ b. Report has been sent to responsible AEC patent group for clearance.

□ c. Patent clearance not required.

4. IMAGE PROCESSING AND PATTERN RECOGNITION RESEARCH: ILLIAC III
(Supported in part by Contract AT(11-1)-2118 with
the U.S. Atomic Energy Commission)

4.1 INTRODUCTION

High lights of the work this past quarter include the following:

(1) Interactive picture processing using the Show-and-Tell System is now operable, if not complete. The current status of the Show-and-Tell System, the Scan-display Device status and Inter-machine Link is described in section 2.

(2) Signal Detection Theory is now being applied successfully to the local characterization of pictures. Extensions of this work called "Interval Coverings" promises significant contributions to scene segmentation. This work is described in section 3.

(3) The first half of Iterative Array of the Pattern Articulation Unit of Illiac III is now operational. The lights (all 512 of them) are "on again all over the world." See section 4.5.3.

(4) Image processing has been applied to Pap smears with limited success. See section 4.5.3

We are beginning to take seriously the responsibilities for maintaining the PAX II picture processing language on the IBM 360/75. Those who would like to join in establishing the PAX II Users Group should contact Val G. Tareski, ILLIAC III Project, Department of Computer Science, University of Illinois, Urbana, Illinois 61801.

B. H. McCormick

4.2 PICTURE PROCESSING STRATEGIES

4.2.1 Show-and-Tell

4.2.1.1 System Objectives

The Show-and-Tell (S&T) programming system is a console-oriented software package providing a facile method of using the operational components of the Illiac III computer. The system operates on the hardware depicted in figure 1. Overall documentation of the Show-and-Tell system is contained in reference [1].

S&T is designed to support local image processing and image acquisition directly using the Illiac III PAU and S-M-V Systems, and also to support experimentation in image processing theory using the high-level language and mass storage of the IBM 360/75. To this end, a bidirectional link with the IBM 360 is provided in the form of a means of calling IBM 360 programs and subroutines from the PDP/8e console teletype, and a set of subroutines for use by IBM 360 programs in transmitting data and pictures back and forth between the PDP/8e and the IBM 360.

Due to the presently ill-defined requirements for picture-processing software, it was deemed wise to implement a minimal system at first, expecting that additions and changes would naturally occur as work on applications progresses. To this end, the system is fairly modular and includes provision for loading parts of the system into the very restricted (8K) PDP/8e core as they are required so as to reduce the need for tight or highly finished code.

4.2.1.2 System Elements

Show-and-Tell is currently composed of the following PDP/8e resident subsystems and IBM 360 resident subsystems:

PDP/8e resident subsystems:

- Translator: Converts S&T language typed by the operator into an interpretable list structure. Enough information is contained in this internal coding to reconstruct the source statements on demand for listing and editing.

[1] Read, John S. DCS Report No. 429, "Show-and-Tell System Specifications," University of Illinois, Urbana, Illinois, March, 1971.

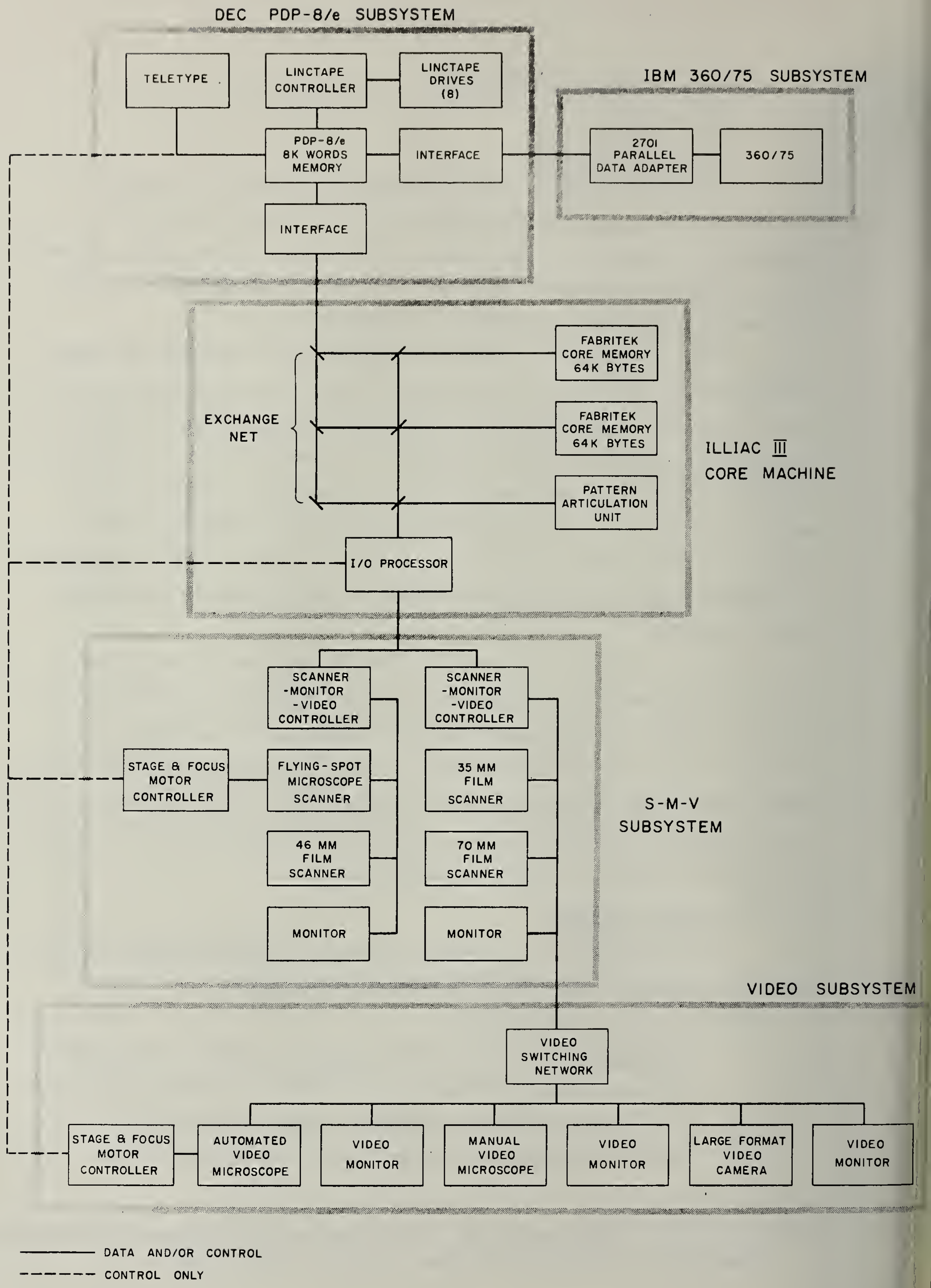


FIGURE 1. HARDWARE CONFIGURATION

- Interpreter: Operates on the aforementioned list structure and on the data storage lists.
- Interpretable Code Area (ICA): Storage for the list structure produced by the Translator.
- Executable Code Area (ECA): Storage for programs executed directly by the PDP/8e hardware; i.e. the output of the PAL assembler.
- Interpretable Code Loader: Loads interpretable code into the ICA.
- EC Loader: Loads executable code into the ECA.
- Supervisor: Provides first-level interrupts handling and controls loading of System Command Processors.
- System Command Processors: Carries out system commands.

IBM 360 resident subsystems:

- PAXDRIVR: Accepts commands and subroutine calls from the PDP/8e, reformats argument strings for FORTRAN compatibility, executes subroutines.
- S&T/360 subroutine package: FORTRAN-callable subroutines to permit the IBM 360 to perform exchange of data with the PDP/8e. Also included are some S&T compatible versions of PAX II subroutines.

4.2.1.3 Documentation and Status

Progress is reported below under the subsystem headings described in the previous section. The reader is referred to reference [1] for a more complete picture of the context of this quarter's work.

Translator: A new version of the Translater was written and debugged. The new version is more systematic than the first one, and the addition of new functions is expected to be much easier. As of April, 1971, the following are acceptable to the Translator:

Labels

GOTO

CALL

INCR

DECR

ASGN

IFGO

EXIT

SCANM

SHOW

LOOK

Coordinate references (direct and indirect)

Integer references (direct and indirect)

Picture references (direct and indirect)

Relations

Signed integers

and Text literal (of length $\leq$ 6 characters)

Interpreter:

The following instructions are now obeyed by the Interpreter, with restrictions as noted:

GOTO

CALL (to subroutine PAX_8 only)

EXIT

SCANM

SHOW

LOOK

A built-in S&T subroutine, PAX_8, has been implemented to handle the interface with the IBM 360 resident subsystems. Functions currently supported are:

Send a subroutine call to the 360

Send a picture to the IBM 360 from Illiac III core

Receive a picture from the IBM 360 and place it in Illiac III core

Type an error message from the IBM 360

Receive and carry out a command from the IBM 360 to scan a picture with Illiac III core

Receive and carry out a command from the IBM 360 to display a picture residing in Illiac III core.

Send a command to the IBM 360 to delete the IBM 360 resident task.

Interpretable Code Loader and Executable Code Loader:

Only a small amount of time was spent this quarter in considering procedures for paging code in and out of the PDP/8e's 8K memory. This facility has not yet become necessary, although it is clear that it soon will.

Supervisor: A table-driven Supervisor has been implemented which handles TTY interrupts and routes control characters [such as those which control cursor movement] to the proper routines. The Supervisor also interprets system commands and calls the proper processor. The current version assumes that all the required control character routines and command processors are in core.

System Command Processors: The command processors listed below were operational by the end of this quarter:

GO -Execute the program in the Interpretable Code Area

KL -Terminate the IBM 360 task comprised of the IBM 360 resident subsystems

LD -This function is being handled temporarily by the DEC system program loading function

SV -This function is being handled temporarily by the DEC system program saving function

WK -Wait for IBM 360 task to begin

XG -Translate and execute each statement as it is entered

XL -Translate statements and store in Interpretable Code Area for later execution.

PAXDRIVR: By the end of the quarter, PAXDRIVR was able to call a PAX II subroutine (or any subroutine following OS/360 conventions), and terminate itself on command from the PDP/8e. Legal subroutine arguments are text strings, integers and references to plane or stacks of planes.

S&T/360 Subroutine package: All of the subroutines listed in section 4.3.2 of reference [1] are implemented. Also, some new subroutines were added this quarter. Fuller documentation on the latter will be issued in the near future.

New Subroutines:

DISPLY	-Send a picture to the Illiac III and display it
SAVPIC	-Write picture on IBM 360 disk or tape
SHOW	-Display current contents of Illiac III core on monitor
XRANPI	-Generate a pseudo-random binary picture
XPRINT	-Print picture on IBM 360 printer
XPUNCH	-Punch picture on IBM 360 card punch.

John S. Read

4.2.2 Scene Segmentation

We have found a useful conceptualization of scene segmentation strategies as an extension of classical clustering techniques.

The levels of generalization allowed in this conceptualization may be described by three classes: classical unirelational clusterings, multirelational clustering, and relational inference.

Classical clustering techniques operate on a fixed graph whose elements are related by a symmetric similiarity measure. Many current scene segmentation strategies are in this class.

Multirelational clustering allows different relations (e.g. colors) on a graph. The procedures can be visualized and implemented as the iterative application of basic graph transformations.

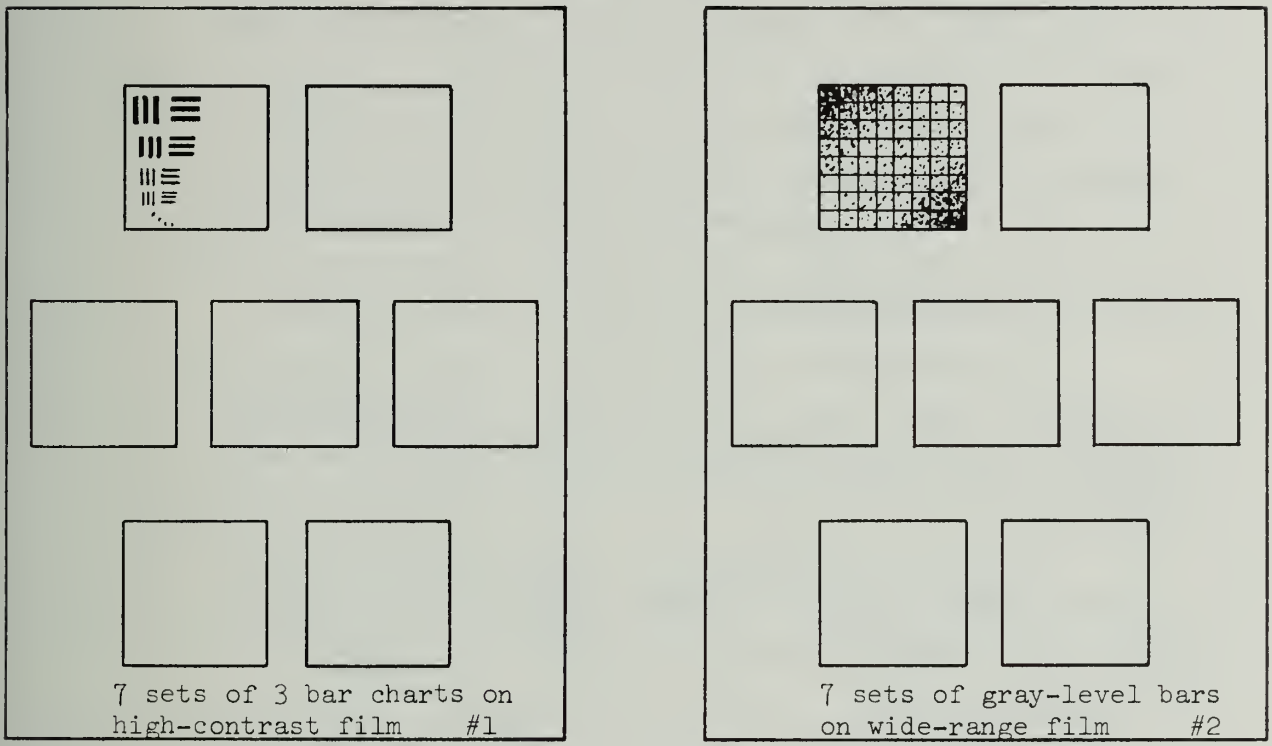
Relational inference allows us to form composite elements over a graph which has unspecified relations. The inference of unknown relations makes it plausible to generalize and find simplifying relational coverings.

Work is proceeding to express our previous results in this unifying framework.

John C. Schwebel

4.2.3 Scanner/Monitor Evaluation

PDP/8 machine language and FORTRAN programs are currently being written to calculate the performance parameters. Also, a revision has been made to standard film frames #1 and #2 to better compromise between efficiency and accuracy.


7 sets of 3 bar charts on high-contrast film #1
7 sets of gray-level bars on wide-range film #2

New Standard Film Frames

Walter Donovan

4.2.4 Scan-Display Devices

Final alignment was completed on the automatic video microscope. All drive motor problems have been solved--the motors are under manual control until the PDP-8e arrives to supply program control. A repeatability check showed that the X and Y axis motors relocate the image ± .0002 inches reliably. A 200 step per turn motor will replace the current 24 step per turn unit for Z axis (focus) control.

The transport section of the flying spot automatic microscope has been moved to the outboard section of the 46 mm scanner frame. It is being readied to operate under the current 46 mm control (alternately with the current film scanner) in order to get the microscope scanner operating as soon as possible.

A second scanner control unit is being built-up in the 70 mm scanner frame. When it is operational, either the new controller or the current 46 mm unit will be altered for television digitizing.

One 46 mm film transport is being modified to take 35 mm sprocketed film, as many of the image samples being investigated are in this form. Losses in quality are incurred when the 35 mm film is reformatted to 46 mm.

A preliminary set of reference images for the scanners has been outlined by Donovan. These are being argued out and a standard set should be agreed upon and manufactured by the end of next quarter. There will be a set for each format (including high resolution television) which will allow systematic checks on resolution, distortion, and gray-scale parameters for each input device. Hopefully, these will eliminate qualitative bickering among users and engineers.

R. Amendola

4.2.5 SMV Controller

Four additional sections of the SMV manual are complete. Relevant logic diagrams and flow charts are complete to this point.

Work on construction of the second SMV Controller and the video-digitization circuitry was slowed this quarter in favor of putting emphasis on adding a microscope scanner to the existing controller. This was done to facilitate earlier testing of the rapidly-developing cytological image-processing work. [See Section 4.5].

Rough wiring to the microscope scanner is almost complete at this time (March) and will be completed early next month.

John Read
J. V. Wenta

4.2.6 Intermachine Link

While awaiting arrival of the PDP/8e work has continued on the definition of various interfaces. Among those nearing completion are:

a) Interface to 360 via 2701 PDA: This is essentially an update of a similar interface constructed by AEC contract 1469 (Professor Gear). See DCS Report 372. This device requires a KD8/E single-cycle data break interface.

b) Maintenance Processor/Exchange Net Interface: This is a flexible, high-speed interface to the Exchange Net which is capable of accessing any units in a variety of modes. Operation requires a KD8/E single-cycle data break interface.

c) Real Time Clock: This device is a crystal-controlled clock suitable for use as an interval timer. The clock features a program selectable time base.

d) Bus Drivers and Device Selectors: These are essentially copies of devices designed by contract 1469 to facilitate interfacing for programmed I/O devices. See DCS File No.842.

Approximately 12 cards have been designed, wire-listed and are ready for wiring.

Richard T. Borovec

4.2.7 Low-Speed Terminal Network

Work in this area has been primarily directed toward resurrecting the LINC tape transports on hand. A transport has been operated under manual control with the most apparent problem being excessive noise in the rear bearings of the transport motors.

Some tests have been devised to check the condition of the transport heads and of the associated read/write electronics. In this regard, Charles E. Molnar of Washington University, St. Louis, was kind enough to mark several tapes on a PDP-12 adjusted for an absolute minimum of head skew. These tapes will allow proper adjustment of our magnetics.

Richard T. Borovec

4.3 PARALLEL PROCESSING STRATEGIES

4.3.1 Synthesis of Interval Coverings for Pattern Recognition:

In a discrete vector space E are given two subsets E^1 and E^0. The subset E^1 consists of vectors (events) measured from a distinguished class of signals (e.g. texture of a picture segment, border between two picture segments, etc.) and the subset $E^0$of vectors from the background. In general the intersection $E^{\phi} = E^1 \Omega E^0$ is a non-empty set; and in this case we order the elements of E^{ϕ} with regard to ratio of frequencies of its occurrence in E^1 and E^0, respectively.

The question considered is how to find an ordered set of filters in the form of multidimensional intervals for recognizing events from the signal class such that the individual filters correspond to the consecutive points on the optimal receiver-operating characteristic defined as in statistical decision theory. This class of filters can be viewed as an ordered covering of the subset E^1. A synthesis procedure for constructing a quasi-optimal ordered covering, using the method of disjoint stars, is being prepared as a report and for conference talks.

B. H. McCormick
R. S. Michalski

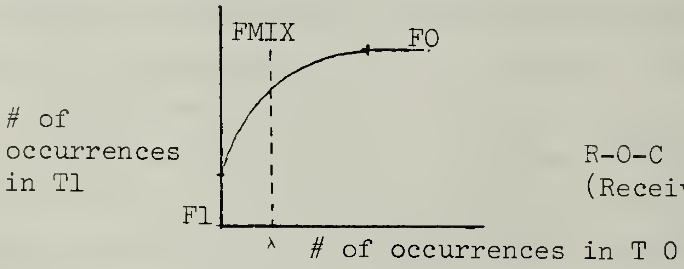
4.3.2 Signal Detection Theory

Regions of different textures can be differentiated from each other if it is possible to extract the patterns which characterize these textures. Ideally each texture consists of only one single pattern, known as 'texture element' (unit cell). But extracting the 'texture element' (whose shape and size is generally unknown), in a natural texture is a formidable task. Instead we can define a complete set of patterns and search for the occurrence of these patterns in the given textures.

We select a 2 x 2 window with 4 gray levels which defines 256 patterns. Each different pattern we call an event.

Given two textures T 1 and T 0, we scan these areas to extract 'events' and divide them into three disjoint sets F1, FMIX, and FO. F1 and FO consist of events which have occurred exclusively in T 1 or T 0, whereas FMIX consists of events which have occurred in both of these areas ordered according to the likelihood ratio (# of occurrences in T1/ # of occurrences in T 0).

By selecting a suitable λ we can extract a set E of events which are more likely to occur in T 1 as compared with T 0.



R-O-C
(Receiver Operating Characterist

Extraction of different textural regions in a given scene:

Given n proto-samples of textures ($t_1, t_2, \ldots, t_n$) which might occur in a scene, we will extract n sets of events $E_1, E_2, \ldots, E_n$ using the R-O-C technique by successively comparing t_i with union of other samples ($\cup\, t_j$) $j \neq i$ for all $i = 1, \ldots, n$.

After obtaining E_i, we extract events from different spatial regions in a given scene of analysis and mark those regions accordingly depending on to which particular set E_i the events belong.

Satisfactory results were obtained when this technique was applied to mark the nucleus, cytoplasm and background in a picture of brain cells.

S.N. Jayaramamurthy

4.3.3 Global Image Straightening

Fortran IV coding of the preliminary version of the curvilinear transformation algorithm was continued, but it has not yet been completed since other tasks dominated.

V. Tareski

4.3.4 Pattern Articulation Unit

4.3.4.1 Iterative Array

The principal work on the IA this past quarter has been to bring section 3 (one-half of the IA) to the level of section 1 (the currently operating half of the IA). All wiring has been completed except for the control/border wiring and the IA/TM interconnections.

4.3.4.2 Control

The design of the Phase 0 has been completed. Also the instruction TOPOLOGY has been implemented.

The GATEIA instruction has been operational during the past quarter. The results of tests indicate that a PLANE WRITE signal of about 500 ns. is sufficient to insure proper operation and therefore, a GOP (Gate Operation) time of 1 μ sec. is reasonable.

Since all the "dirty details of implementation" (i.e. fan-in and fan-out logic, etc.)have been completed, design will continue on an instruction by instruction basis. Approximately 15 instructions are flow charted awaiting implementation.

R. T. Borovec

4.3.5 PAX II Language Support

The following additions and changes have been made to the IBM 360 version of the PAX II Picture Processing System: two new FORTRAN subroutines, DCONV and LBLNUM, and one new assembly language routine, DATE, have been added. IBM 360 versions of the following UNIVAC 1108 PAX Subroutines have been added: ABSUBT, CHECKW, FINWD, GLHS, GRIDW, LABEL, LINE, LSTCOM, MULT, MULTC, NAND, NANDS, NOR, NORS, PTHS, ROT9Ø, SETSIZ, SUBSIN, and XREFL. Errors in FORTRAN subroutine PRINT and assembly language routine FP were corrected. Subroutines READZ, WRITEP, and WRITEZ were changed to utilize routine KPBS. Assembly language routines KGC and KPC were rewritten to reduce their execution times.

One more "PAX Memo" has been sent to University of Illinois PAX users to describe the latest version of the PAX II Subroutine Library, PAXSLIB. A full description of IBM 360 PAX memory requirements was included, as was documentation describing the new and added routines.

The first issue of the "PAX Users Group Newsletter" has been sent to all installations known to have a version of the PAX II Subroutine Package. Requests for copies of this newsletter should be sent to Pax Newsletter Editor, Illiac III Project, Department of Computer Science, University of Illinois, Urbana, Illinois 61801.

V. Tareski

4.4 GRAPH TRANSFORMATION STRATEGIES

4.4.1 Graph Transformational Languages

We have specified a Root Graph Language, RGL, in order to be able to simply and conveniently express operations on graph networks which allow an arbitrary number of associated variables with any network element.

To justify the definition of yet another "programming language" and put its development into perspective, we cite the following facts: We are currently able to program picture processing algorithms which operate on the PAX plane representation of pictures as binary valued elements with neighborhood connectivity relationships. The next and most natural abstraction from a PAX plane representation is a graph network. Many scene segmentation algorithms can be expressed most simply by operations on a graph network. We have developed some theory for the next higher level of picture processing operations represented as graph network transformations.

Thus, the graph language is quite helpful for precisely specifying and for experimenting with heuristic scene segmentation strategies.

The language has a small number of root operations which are sufficient for expressing network transformations and was specified with the goal of embedding it in PL/I.

J. Schwebel

4.4.2 Taxicrinic Processor

Work has continued on Volume II of the TP Manual. It should be ready to publish early in the next quarter.

The card design and wiring of the integrated circuit portions of the TP control logic and register sections has progressed well. At the present time, about 102 final drawings have been completed representing about 70 IC boards. This does not include the inter-board connections. Of the boards whose drawings have been completed, about 57 have returned from the wiring contractor and the rest are in the process of being wired. Physical checkout of the boards has just been started.

During this quarter the first two volumes of the new edition of the Illiac III Programming Manual were completed. These two volumes have included the description of the computer system and the instruction repertoire. Volumes III and IV, which are currently in progress will include the input/output operations and the supervisor organization.

The logical simulation of the Taxicrinic Processor continued during the first part of this quarter. However, the size and complexity of the simulator has become such that it greatly taxes the capacity of the Digital Computer Laboratory's IBM 360 system as well as the single programmer working on it. As a result, work on the simulator was halted after about 1/3 of the basic machine control logic had been simulated. The sequences which have been programmed operate successfully. However, due to the various system problems which come up whenever a single program needs an exceptional proportion of the total system resources, it was felt that the time spent on the simulator could be more usefully spent in other phases of the taxicrinic processor design.

B. J. Nordmann, Jr.

4.5 APPLICATIONS

4.5.1 SEM Micrographs

The Scanning Electron Microscope has been extensively employed in its applications to material science as well as subcellular biology. A JSM-3 SEM, Japan Electron Optics Laboratory Company was installed in the Material Research Laboratory, University of Illinois, Urbana, late last summer. Mr. John B. Woodhouse of MRL is in charge of the facility. With his help, a program to explore the benefits of image processing automation for SEM imagery was initiated. The system design of an automated on-line scanning electron microscope has been in process.

One of our main goals is to study cells and subcellular populations as generated by the cytospectrometer (see Section 4.5.2 below). We have undertaken two preliminary tasks:

(1) Slide preparation: With the services of Mrs. Beth Lepinski and Mrs. Grace Conway of Mercy Hospital, Urbana, Illinois, we have prepared a variety of samples such as red cells, leukocytes, urine crystals, a few kinds of bacteria in different forms, chromosomes, etc., courtesy of Dr. Ben Williams.

(2) Evaporated coating techniques: The problem arises from preparation of a slide to be available to be observed both under conventional optical microscope and SEM. With the help and suggestions of Professor Francis Young of Civil Engineering Department, University of Illinois, Urbana, a thin coating of 200 Å of carbon and $Au/_{Pd}$ alloy is a compromise to solve the problem. For a further improvement, we might try to treat the sample with OsO_4 vapor* to insure good image quality.

J. Chen

* Gerhard E. Pfefferkorn, "Specimen Preparation Techniques," Proceedings of 3rd. Annual Scanning Electron Microscope Symposium, 1970, p.91

4.5.2 Cytospectrometer

The main effort this quarter has been the design, construction and checkout of the droplet generator digital control. A brief description of the control follows.

A pulse source, currently an audio oscillator, is used as a master clock. The clock drives a switch-programmable frequency divider chain. The divider consists of a twelve-bit counter and associated reset logic.

The output pulse of this divider is used to drive both a high-voltage charge ring driver and a stroboscope driver. This pulse also drives a switch-programmable frequency divider chain consisting of an eight-bit counter.

The output pulse of this divider is used to drive the high voltage deflection plate drivers.

Independent adjustment of pulse width and delay are provided for the charge ring, strobe, and deflection plates.

The drivers for the charge ring and deflection plates are simple, single transistor stages. Output voltages may be set between 0 and ± 350 volts. The outputs are current-limited to reduce the possibility of serious electrical shocks.

The bimorph driver is also a single transistor device. Since the bimorph exhibits very large capacitance at high frequency, operation of this prototype driver is limited to about 40 KHz. The bimorph driving signal may be varied from 0 to 60 volts.

The entire control, less power supplies, is contained in a 10 3/4" rack panel. The twenty-five integrated circuits are mounted on an 8 x 10 inch wire wrap board to facilitate easy modifications.

R. T. Borovec

4.5.3 Pap Smears

To make automated or semi-automated measurements on large numbers of cells presented in the standard Pap-smear format, it is necessary to have an efficient searching procedure which can reliably locate cells and cell components in the presence of pictorial noise caused by cell clumping and overlapping.

This quarter, a Pap-smear searching program using Illiac III-type instructions was designed and partly implemented. The design and some preliminary results were presented at a biomathematics symposium in March. [see Section 4.7 this report]

The search procedures are based on the Illiac III's plane-parallel operations and were programmed and tested using the PAX II language interactively in conjunction with the Show-and-Tell programming system [see Section 4.2.1, this report].

Figure 1 is a microscope field showing a portion of a Pap smear magnified by a factor of 63 (on the 35-mm film). The currently-implemented routines operate on a coarse-resolution scan, as in figure 2, [128 x 128 pixels, 16 gray levels covering a field of approximately 225μ x 350μ on the specimen]. The programs label points in the picture as cell boundary, cell nucleus, leukocytes, background and clump. The label for each point is generated with parallel, Pattern Articulation Unit-type operations and stored in a plane or "map" having a bit set on for each picture point having the label. A given point may have more than one label; i.e., may be "on" in more than one plane.

Examples demonstrating this labelling are in figures 3 and 4. Figure 3 shows lines drawn by the program around points identified as clump points, that is, places where a texture was detected which indicated that a leukocyte clump might be present, but where individual cells could not be resolved. In figure 4, the program has removed points labelled as likely to be leukocytes. The "holes" were filled with background gray-scale values. Note that these displays were generated primarily to

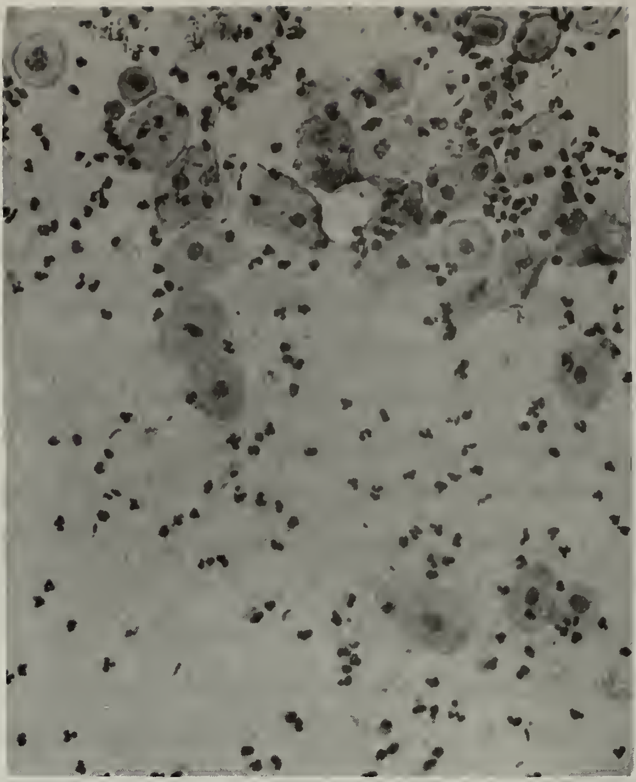


Figure 1. Microscope view of Pap smear

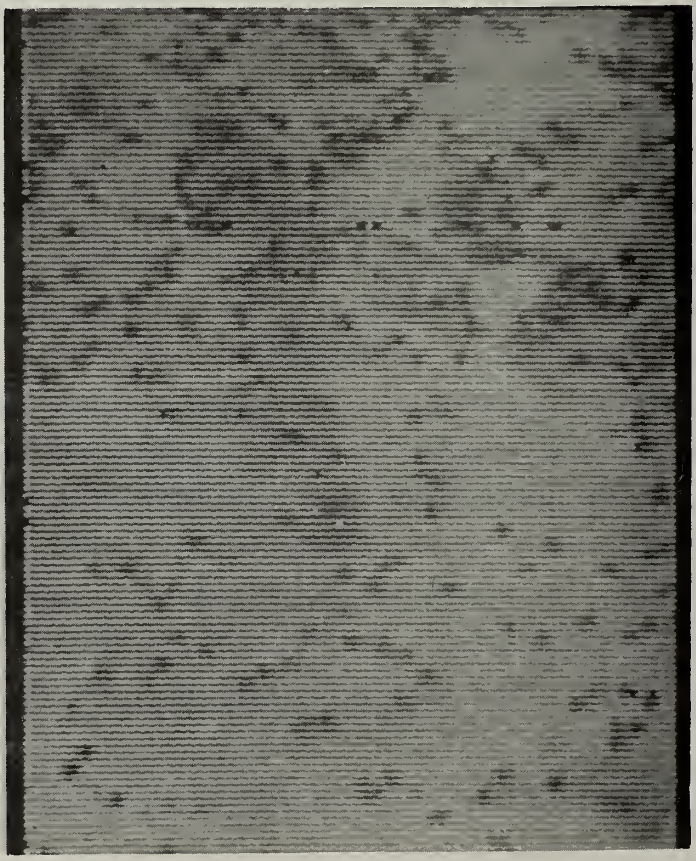


Figure 2. Digitization of figure 1 (coarse resolution)

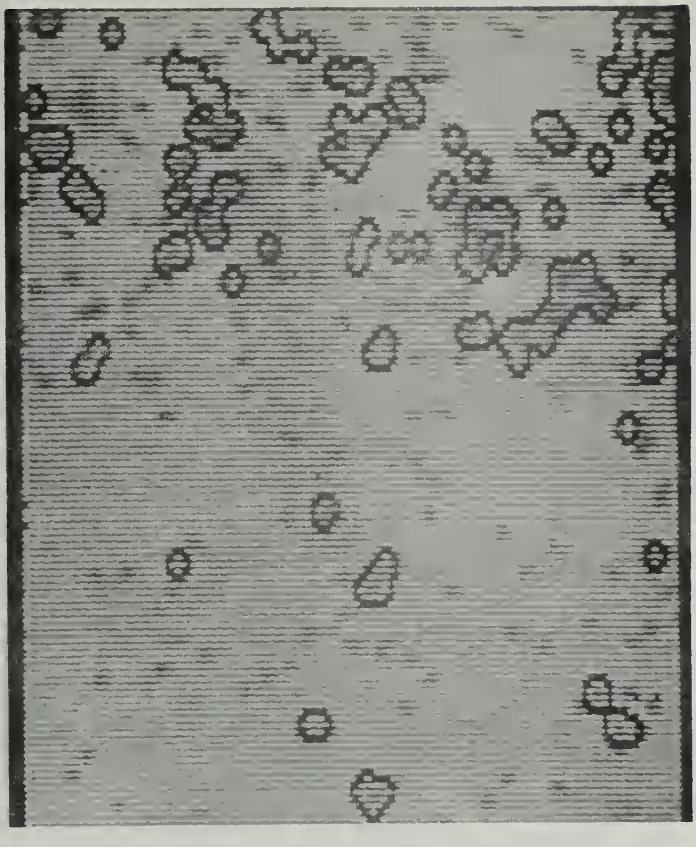


Figure 3. Clump points labeled

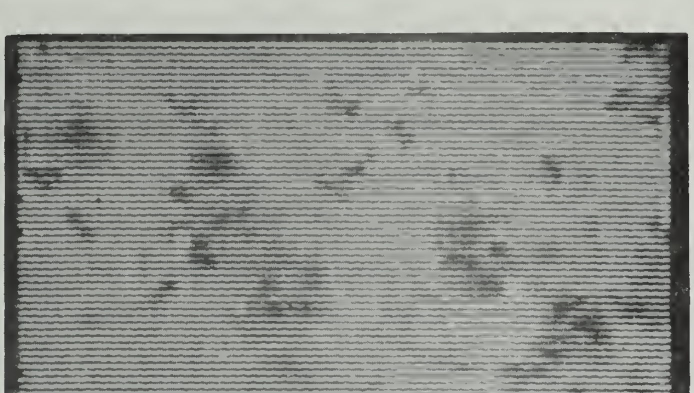


Figure 4. Leukocytes identified and removed

permit tracing of the program's progress. These pictures are not used per se in the processing. Information required in this way is used to direct rescanning of small areas at higher resolution, thereby pruning the search tree relatively rapidly. When objects are found which are known to be ambiguous at the coarse resolution, they can be re-examined with specific procedures to resolve specific ambiguities.

John S. Read

4.5.4 Brain Mapping

The purpose of this investigation is to develop inference techniques and to test and apply them in the environment of neuro-anatomy of the brain.

It is intended to apply these techniques first to a small subset of objects in the brain. In connection with current research performed in quantitative neuroanatomy*, methods are being developed for the automatic recognition of cellular and subcellular objects visible in cresylecht violet stain and Weil stain as far as they can be resolved under a light microscope. An immediate application is to determine quantitative properties such as the number of neurons in their specific domains of the brain and the geometrical dimensions of their nuclei and nucleoli. A further application is the delineation of the boundaries of larger domains in the brain, e.g. nuclei, such that finally a brain map is obtained.

Our approach consists of a preprocessing and an inference phase. In the preprocessing phase, the picture is scanned (i.e. transformed into a grid of points differing in gray scale) and partitioned into closed regions. Two methods to obtain regions are presently being investigated. A region is determined as a connected set of grid points such that (1) they have almost uniform distribution of gray scale values or (2) they form a more complicated characteristic texture. After

Fry, William J., "Quantitative Delineation of the Efferent Anatomy of the Medial Mammillary Nucleus of the Cat," J. Comp. Neur. V. 139, No. 3, July 1970, pp. 321-336 partitioning a picture into regions, we associate with each region applicable attributes and their values as well as binary relations valid between pairs of regions.

Region identifiers, attribute values, and binary relations constitute the input to the inference phase. Two strategies are being considered:

1. A simple context-free grammar has been developed whose nonterminal values are class names used in neuroanatomy to designate objects such as cell-types, constituents of cells, etc. The variables are associated with attributes that are connected by semantic rules. Using clustering and covering techniques that are already available or being developed, a variable of grammar is assigned to each region such that the structure of the picture is described by a derivation tree whose nodes are associated with a region identified.
2. To generate appropriate grammars using covering and clustering techniques.

It is intended to make use of all clustering techniques that are available. The great diversity of objects constituting a brain (gross anatomy, histology, neural microanatomy, etc.) allows us to select specific methods of varying degrees of sophistication. The following techniques are being considered:

1. Graph theoretical clustering methods (see Section 4.4.1)
2. Covering techniques that are being developed in the framework of signal detection theory (see Section 4.3.1)
3. Probabilistic clustering
4. Grammatical inference

Peter Raulefs

4.6 COMPUTER SYSTEMS SUPPORT

4.6.1 IBAL Assembler

IBAL (Illiac III Basic Assembly Language) grammar was reviewed. CF productions were partly incomplete. Grammar for integer expressions and boolean expressions were modified to a PL/1-like structure. Segmentation and synchronization, which are important in file management, memory allocation and parallel processing, are still under consideration.

The work to write the parcing phase of the assembler, using Alan James Beal's "Translation Writing System," has started. The grammar has to be rewritten in TWINKLE which is an English-like-meta-language for writing grammars.

A revised IBAL Manual will be issued early in the next quarter.

A. Masumi

4.6.2 Operating System

The overall framework of the Illiac III Operating System, specifically the part which will be implemented in hardware, is almost completed. The necessity of knowing more about the environment in which this OS will be functioning has made us work in parallel on the assembler system and on the file management system which will be an integral part of Illiac III OS.

The consideration of IBAL as a general assembly system is leading to some programming convention for our system, which in turn will specify environmental requirements on the operating system.

A. Masumi

4.6.3 Arithmetic Units

Control logic drawings for the instructions: Add, Subtract, Compare Algebraic and Multiply were completed and proofed. Layout on PCB's is still being done and should be completed in the coming quarter. Drawings for Division instruction were completed and are being proofed. Cross-referencing between AU processing hardware drawings is being continued.

L. N. Goyal

4.6.4 I/O Processor

The System Organization of the I/O Processor was delineated in Referance Manual, Vol. I* issued this quarter. Vol. III which discusses in considerable detail the Command Repertoire of this processor, should be available for printing in the next quarter.

J. V. Wenta

4.6.5 Channel Interface Units

The wire count for the integrated circuit cards was completed.

Duplicate wire lists for these cards have been completed and corrected and made available to the vendor. Vendor has completed the wiring of one complete set of IC cards and promised delivery on April 2, 1971.

Listing of flow charts and completed logic diagrams for CIU manual is complete.

J. V. Wenta

4.6.6 Device Controller

There have been no new developments on this item during this quarter.

Val Tareski

4.6.7 Diagnostic Procedures:

The study undertaken on the subject "Control Point Strategy and Its Automatic Diagnosis" (Master Thesis to be published) is now in its final phase. During the last months, two points have been

*McCormick, B.H., Nordmann, B.J., "ILLIAC III REFERENCE MANUAL, VOL. I: The Computer System," DCS Report No. 433, University of Illinois, Urbana, Illinois, February 17, 1971.

given particular attention: the development of a generator of fault detection tests for the control cards and the design of the Sequence Tester.

At the present time, the generator has been completed. A program called "CPTEST" has been written in FORTRAN which implements two "merging algorithms" developed in the above thesis. The idea is:

(1) to functionally divide the card into subnetworks

(2) to generate tests for these subnetworks

(3) to merge the obtained tests.

The structure of the Sequence Tester, interface between the card and the PDP/8e which monitors the testing process, is developed in the above thesis. Its design (packaging, wiring lists, etc.) is 80% completed and its construction has begun.

C. Rey

4.7 DOCUMENTATION

4.7.1 External Documents Issued

Report No. 434 and 433 entries:

Report No. 433 — ILLIAC III REFERENCE MANUAL, VOL I: The Computer System, edited by B. H. McCormick and B. J. Nordmann, Jr. February 17, 1971

Report No. 434 — ILLIAC III REFERENCE MANUAL, VOL II: Instruction Repertoire, edited by B. H. McCormick and B. J. Nordmann, Jr., February 26, 1971

Outside Talks

Read, John S., "Image Segmentation Techniques for Automated Cervical Smear Processing," 9th Annual Symposium on Mathematics and Computer Science in the Life Sciences, Anderson Hospital and Tumor Institute, Houston, Texas, March 22-24, 1971.

Seminars: Sight Sensory Systems

"The Synthesis of Interval Coverings for Picture Filtering," I, II, III, IV, Professor R. S. Michalski, Department of Computer Science, University of Illinois, January 14, 21, February 17, 24, 1971

"Quantitative Neuroanatomy," Professor Frank Fry, Department of Electrical Engineering, University of Illinois, March 3, 1971

"Investigations in Artificial Non-Symbolic Cognition," Professor Sylvan Ray, Department of Computer Science, University of Illinois, March 10, 1971

"Image Segmentation Techniques for Automated Cervical Smear Processing," John S. Read, Department of Computer Science, University of Illinois, March 17, 1971

"Scene Segmentation with a Graph Transformational Model," John C. Schwebel, Department of Computer Science, University of Illinois, March 24, 1971

"The Particle Transport System: Its features and foibles," Robert C. Amcndola, Department of Computer Science, University of Illinois, March 31, 1971

4.7.2 Logic Drawings Issued

37 TP and PAU control logic and 39 TP processing hardware logic drawings have been drawn and issued during the last quarter.

SMV control logic drawings have been updated to current design.

All documentation of control logic is proceeding in parallel with the design.

Set of (23) Sequence Tester logic drawings has been submitted to drafting and is presently being processed.

4.7.3 Engineering Drafting Report

During the last quarter a total of 329 drawings, including drawing changes, layouts, flow charts, thesis, report drawings and drawings related to Opto/Mechanical design of Illiac III have been processed by AT(11-1)-2118 drafting section.

S. Zundo

4.8 ADMINISTRATION

4.8.1 Personnel Report

Senior Staff

Professor Bruce H. McCormick - Principal Investigator

Assistant Professor - R. S. Michalski

Professional Staff

Robert C. Amendola

Richard T. Borovec

John S. Read

Research Engineering Assistant

S. Paul Krabbe

Electronic Engineering Assistant

Joseph V. Wenta

Digital Computer Technician II

George T. Lewis

Drafting

Stanislavs Zundo

Research Assistants

Jerry Chen

Walter Donovan

Lakshmi N. Goyal

Richard P. Harms

S. M. Jayaramamurthy

Ahmad Masumi

Bernard J. Nordmann

Christian Rey

John C. Schwebel

Val G. Tareski

Secretarial

Mrs. Roberta Andre'

4.8.2 Computer Usage Log Summaries

Scanner-Monitor-Video:

During the past quarter, the S-M-V was under power approximately 175 hours with the following division of usage:

Operational	128 hours
Preventive Maintenance and Testing	24 hours
Corrective Maintenance	23 hours

IBM 360/75

Total first quarter expenditure was $1,713.93

John S. Read

Form AEC-427
(6/68)
AECM 3201

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.
COO-2118-0007

2. TITLE
QUARTERLY PROGRESS REPORT - JANUARY-MARCH, 1971

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report
☐ b. Conference paper not to be published in a journal:
Title of conference ______
Date of conference ______
Exact location of conference ______
Sponsoring organization ______
☐ c. Other (Specify) ______

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.
☐ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
☐ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)
Professor Bruce H. McCormick
Principal Investigator
Illiac III Project

Organization
Department of Computer Science
University of Illinois
Urbana, Illinois

Signature Bruce H. McCormick

Date April 15, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

☐ a. AEC patent clearance has been granted by responsible AEC patent group.
☐ b. Report has been sent to responsible AEC patent group for clearance.
☐ c. Patent clearance not required.

5. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES

(Supported in part by the National Science Foundation under Grant No. US NSF GJ-812.)

5.1 Computerized Mathematics

We continued our research on the naming rule as an additional rule of inference with resolution. The naming rule corresponds to the axiom $C(x, y_1, \ldots y_m) \longleftrightarrow \epsilon(d(y_1, \ldots y_m), x)$ (or $C(z, x, y_1, \ldots y_m) \longleftrightarrow \neg \epsilon(z, d(y_1, \ldots y_m), x)$ where C is a negative clause with all ternary relation symbols). These axioms, intuitively, give names to newly constructed predicts or functions. The restrictions on C in the second case are meant to insure that the new object really is a function. The naming rule says that, given the clause C, one can deduce $C(x, d(y_1, \ldots y_m))$ or $\neg \epsilon(z, d(y_1, \ldots y_m), x)$. Conversely, the unnaming rule states that, from $\epsilon(t, d(u_1, \ldots u_m))$ or $\neg \epsilon(s, d(u_1, \ldots u_m), t)$ one can conclude the appropriate instance of C. The need for naming arises from the requirement for Henkin validity--that the interpretation of any formula be in the domain of interpretation. Since a clause can be interpreted as being a predicate or function, we must have a way of guaranteeing that predicate or function is in the domain. This is accomplished by giving that object a name, i.e. $d(y_1, \ldots y_m)$. During this quarter we were able to show that if only one instance of such an axiom was needed for inconsistency, resolution and the naming/unnaming rule is sufficient to generate the empty clause, i.e. together they are complete. More generally, we showed that using resolution alone, one can generate a set of clauses I such that each clause in the set of clause parts of axioms, say C_1, C_2, ... C_k, needed for inconsistency is subsumed by some clause in I. Of course, the set of axioms needed is not known beforehand. However, the above fact asserts that the clause parts of these axioms can essentially be generated by resolution. Thus, in considering possibilities for axioms to add after resolution has failed, one need not consider at one time material from more than one clause.

These results are being collected into a Ph.D. thesis to be presented to the Graduate College for this graduation this June.

(L. J. Henschen)

5.2 Problems In Computational Geometry

An analytical intersection detection procedure for polyhedral objects by means of "even parity modes" has been studied, and its computation results will be shown. As an application of the procedure developed, so called "sofa" problems have been solved and the simulation results for various criteria will be presented.

The concept of a "dynamic" sofa problem will be introduced and an approach for the solution of this problem will be presented.

(K. Maruyama)

5.3 PDP-11

We now have a batch monitor running on the PDP-11. We, also, have interfaced a card reader and line printer to the system. Digital Equipment Corporation has loaned up a high-speed paper tape reader for use until our own Dec tapes are delivered.

We have a basic time-sharing monitor working and hope to have an editor, assembler, etc. for use with it soon.

(D. W. Oxley)

5.4 Factorization Methods Used in the Solution of Partial Differential Equations

We continued to study factorization methods with particular interest given to the symmetric factorization suggested by Stone [1]. The differential equations is approximated by a difference equation at each point of a grid and a system of simultaneous linear equations results.

In matrix notation the system Ax = q is solved using the iteration

$$(A + B)X_{N+1} = (A + B)X_N - \tau_N(AX_N - q) \qquad (1)$$

where B is chosen to make each step of (1) easily computable and the procedure rapidly convergent. We have been concerned with the selection of τ_N which is dependent on the eigenvalues of $(A + B)^{-1}A$.

An algorithm was developed and tested to simultaneously calculate the extreme eigenvalues of $(A + B)^{-1}A$ and the iterate X_N using (1). As the approximation to the eigenvalue converges, a better sequence of τ_N's is selected increasing the rate of convergence of X_N to X.

The algorithm first calculates y_N, an approximate eigenvector of $(A + B)^{-1}A$. Then the approximate eigenvalue

$$u_n = \frac{(Ay_N, y_N)}{((A+B)y_N, y_N)} \qquad (2)$$

is calculated. If y is the actual eigenvector, λ the eigenvalue and $y_N = cy + \sum_{i=2}^{K} \epsilon_i w_i$, where w_i $i = 2, \ldots K$ is an eigenvector of $(A + B)^{-1}A$

orthonormal with respect to A + B and λ_i $i = 2, \ldots K$ corresponding

eigenvalues, then $u_n = \dfrac{\lambda + \sum\limits_{i=2}^{K} \epsilon_i^{2}\lambda_i/C^2}{1 + \sum\limits_{i=2}^{K} \epsilon_i^{2}/C^2}$.

Tests have shown the algorithm works as well as using (1) with a fixed "best" value of τ_N.

For the symmetric factorization suggested by Stone in which B depends on a parameter α, it was shown that for $\alpha = 1$ and A derived from Laplaces equation, $\lambda_{MIN}((A + B)^{-1}A) > 1/2$ and approaches 1/2 as the gridsize h goes to zero.

For $A = (a_{ij})$ with constant diagonals $a_{i,i\pm n} = B < 0$, $a_{i,i\pm 1} = D < 0$ and $\alpha = 1$.

$$\lambda_{MIN}((A + B)^{-1}A) \leq \frac{1}{1 + \dfrac{2\sqrt{BD}}{-(B+D)}} \text{ as } h \to 0.$$

Furthermore if $a_{i,i\pm 1}$, $a_{i,i\pm n}$ are constant on the matrices A_m, $m = 1, \ldots, M$ I.E. $a_{i,i\pm 1} = D_m$, $a_{i,i\pm n} = B_m$ in A_m then

$$\lambda_{MIN}((A + B)^{-1}A) \leq \underset{M=1,\ldots,M}{\mathrm{MIN}} \frac{1}{1 + \dfrac{2\sqrt{B_M D_M}}{-(B_M+D_M)}} \text{ as } h \to 0.$$

It is conjectured that $\lambda_{MAX}((A + B)^{-1}A)$ grows without bound as $h \to 0$. If X is the vector $(X_{1,1}, X_{2,1}, \ldots, X_{n,1}, X_{1,2}, \ldots, X_{n,2}, \ldots X_{n,n})^T$

where $X_{i,j} = \mathrm{Cos}^{20}(3(i-j)h\pi/2)e^{-(1+|ih-1/2|+|2h-1/2|)^2(2-\mathrm{Cos}(3(i-j)h\pi/2))}$

$(n + 1)h = 1$, then for $h = \frac{1}{30}$, $R_h = \frac{(AX,X)}{((A+B)X,X)} \approx 4$

for $h = \frac{1}{100}$, $R_h \approx 10$

and for $h = \frac{1}{250}$, $R_h \approx 18$

$R_h \leq \lambda_{MAX}((A + B)^{-1}A)$ thus it appears that as $h \to 0$ λ_{MAX} grows considerably and the convergence of (1) is slow. Work is in progress to get some theoretical results on the growth of λ_{MAX}.

(M. Diamond)

[1] Stone, H. L., Private Communication, April, 1969.

NOTE: The following work was performed during the period July 1 to October 1, 1970 and is reported on here because it was inadvertantly omitted from the earlier Quarterly Progiess Report.

5.5 Graph Algorithm Research

During this quarter, a new algorithm for finding all the trees in a graph was programmed in GASP and debugged. When tested on the complete graph on nine nodes, the program took under two minutes (on an IBM 360~91), averaging less than 250 nanoseconds per tiee.

The following table compares the computation costs of the various algorithms for finding all trees. "n" is the number of nodes, "b" the number of branches and "t" the number of trees in the given graph. For the complete tree, $t = n^{(n-2)}$. Details will appear in a future departmental report.

Author and Year		Type	Growth Rate of Cost	Complete Graph
MacWilliams	58	exhaustion	$= \binom{b}{n-1}$	$\leq \frac{n^{2n}}{2^n(n-1)!}$
Zobrist	64	exhaustion	$= n^{n-1}$	$= n^{n-1}$
Mayeda	65	tree sections	$\geq t$	$\geq n^{n-2}$
Kamae	67	hamiltonian path	$> t$	$> n^{n-2}$
Ballert	62	cartesian product	$\leq t(\frac{2b}{n})^{n-1}$	$\leq n^{n-2}(n+1)^{n-1}$
Mason	57	cartesian product	$\leq n(\frac{2b}{n})^{n-1}$	$\leq (n+1)^{n-1}$
Feussner	02	$T = T(b)\ \ T(\ b)$	$\geq t$	$\geq n^{n-2}$
Percival	53	factoring	$\leq (n-1)!$	$= (n-1)!$
Chase	70	factoring	$< (.8)^n(n-1)!$	$< (.8)^n(n-1)!$

(S. Chase)

5.6 Stability Charts of Stiffly Stable Methods for Digital Simulation

The stiff differential equations arise in many computer aided design techniques particularly network analysis and simulation. The real-time element in simulation makes the computer operate as in a real word environment where, for example, one second of computer time is one second of actual time. This real-time requirement is particularly important in the training function of simulators, but it is also highly desirable for simulation in general. The real-time digital simulation of continuous systems received its greatest initial impetus in the 1950's when Gray [1] developed the method for constructing the stability charts of numerical integration methods of differential equations.

Gear [2] has developed stiffly stable methods of order as high as six and showed their application to the stiff differential equations. Jain and Srivastara [3, 4] have obtained stiffly stable methods of order K as high as 11 for suitable parameters.

Here we have constructed stability charts for stiffly stable methods. The stability charts enable us to choose an optimal integration step and thus result in saving considerable computer time.

(M. K. Jain)

[1] Gray, J., Jr., "Numerical Methods in Digital Time Simulation," Moore School of Engineering, University of Pennsylvania, Philadelphia, Pennsylvania, (Ph.D. Thesis), June, 1953.

[2] Gear, C. W., "Numerical Integration of Stiff Ordinary Differential Equations," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 221, January, 1967.

[3] Jain, M. K., and V. K. Srivastava, "High Order Stiffly Stable Methods for Ordinary Differential Equations," Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 394, April, 1970.

[4] Jain, M. K., and V. K. Srivastara, "Optimal Stiffly Stable Methods for Ordinary Differential Equations," Department of Computer Science, University of Illinois, Report No. 402, June, 1970.

6. SWITCHING THEORY AND LOGICAL DESIGN

(Supported in part by the National Science Foundation under Grant Number U.S. NSF-GJ-503)

After completing the comparison of the implicite enumeration method (explicit use of inequalities) with the branch-and-bound method (no use of inequalities) and the improvement of the branch-and-bound method, T. Nagakawa left. Y. Kambayaski joined the group. We started to look at the logical design problem with integrated circuit.

S. MUROGA

A program was devised and debugged which will draw (by Calcomp) networks of NOR, OR, AND, WIRED-OR, and WIRED-AND gates when given the connections among gates and external variables. The intent of this program is to prevent time-consuming reconstructions of networks from the connection patterns which are the present form of output from our implicit enumeration and branch-and-bound programs.

(J. Culliney)

Properties of logic families of integrated circuits are studied in order to investigate (1) conditions for wired logic and (2) possibility of mixing different types of logic gates. For example, there exists a NOR gate with wired AND capability (RTL), a NOR gate with wired OR capability (ECL), a NAND gate with wired AND capability (DTL, TTL) and an AND gate with wired OR capability (CTL). Synthesis procedures using logic gates with wired logic capability are not treated before. Several properties of such gates were obtained, which will be used in synthesis methods.

(Y.Kambayashi)

NOR networks with the minimum number of interconnections for all 3-variable functions and some 4-variable functions have been obtained with the program ILLOD(NOR-B). Two out of seventy seven 3-variable functions and five out of fifteen 4-variable functions have the minimum interconnection networks which differ from the networks with the minimum number of gates.

The program ILLOD(NOR-B) is augmented with an optional use of the magnetic tape to store the solutions obtained by this program.

(H.C. Lai)

Some theoretical studies were made on the logic networks which consist of MOS complex cells. A few properties of these networks were obtained. An algorithm was developed for the synthesis of multi-level networks with a minimum number of cells.

(T.K. Liu)

7. SOUPAC SECTION

(Statistically Oriented Users Programming and Consulting)

During the period January through March, 1971, the SOUPAC consultants began the third series of SOUPAC seminars. The weekly seminars have proven a valuable means of communicating the scope and power of SOUPAC as well as giving valuable insights concerning its use. In addition, weekly staff meetings are being held to acquaint members of the SOUPAC staff with the theory and use of some of the newer programs. Some consideration has been given to opening these meetings to the public.

The fourth edition of the SOUPAC Manual was issued in March. This edition contained, for the first time, the programs FIT-Chi Square Goodness of Fit, Kolmogorov-Smirnov D Statistic, Paired Comparisons, Three Mode Factor Analysis, a completely rewritten Transformations, and several revised or additional Matrix operations.

The SOUPAC consultants personally conducted class sessions in SOUPAC for two classes, Advertising 381 and Economics 477. Writeups and manuals were provided for several others.

A program feature called ALLOC8 (allocate) is being incorporated into all programs which are not currently dynamically allocatable with roll-in and roll-out of data. Dynamic allocation with roll-in and roll-out enables a program to be run in a arbitrary region size with arbitrarily large data. This capability generates overhead in execution of a program (notably machine time and I/O) and also requires extensive reprogramming. ALLOC8 enables a program to be run in arbitrary region size, but only on data which can be accomodated by that region size. ALLOC8, by virtue of its simplicity, cuts down on overhead and reprogramming, but maintains the advantage of running in arbitrary region. ALLOC8, in the case of a problem size which is too large for the requested region will not execute but will provide a message indicating how much more region would be required to run the problem. The vast majority of research problems can be fit into a region size less than or equal to the maximum made available to users of our machine. Thus ALLOC8 provides for optimal matching of data to region. Only problems of a magnitude such that they cannot be contained in maximum available region would suffer from the fact that roll-in and roll-out of data is not provided by ALLOC8.

During this last quarter further study of the SPSS statistical package was done. The conclusion of the SOUPAC group was that it was one of the better small statistical packages available.

Kern W. Dickman

8. MATRIX MULTIPLICATION OPERATION MINIMIZATION

Given a product of a sequence of conformable matrices and vectors, a reduction of the number of operations is possible by proper rearrangement of the sequence. That is, since all such multiplications are associative and some are commutative, the time required is highly dependent on the order in which the multiplications are performed. An example is C x R x A where C, R, and A stand for a column vector, a row vector, and a matrix of dimension n respectively. Then the computation (C x R) x A requires $n^3 + n^2$ multiplications whereas an equivalent computation C x (R x A) requires only $2n^2$ multiplications. A simple algorithm which associates pairs of arrays in a way that always gives the minimum number of operations has been developed. It was also shown that the same algorithm could be used to evaluate matrix expressions on a parallel machine with the minimum number of operations as well.

D. Kuck and Y. Muraoka

9. PARALLELISM EXTRACTION

Introduction

A program analyzer which takes a program written in a conventional language, e.g. FORTRAN, and analyzes for a parallel machine (e.g. STAR, Illiac IV) is being programmed in PL/1. The analyzer not only measures potential parallelism in a program but also modifies it if necessary to extract more parallelism. In the past quarter emphasis was put on the analysis of an assignment statement and a DO loop. Parallel computation of an assignment statement has been studied by many people. An important feature being overlooked is the possibility of reducing computation time by distribution, e.g. compare parallel computation of two equivalent expressions $\sum_{i=0}^{n} a_i x^i$ and $a_0 + x(a_1 + x(a_2 + \ldots(a_{n-1} + a_n x)\ldots)$. An algorithm has been developed and implemented to achieve this end.

The DO loop analyzer has also been coded. Given, e.g. a matrix addition coded in FORTRAN, the analyzer indicates that it can be computed in one step on a parallel machine, that is, element-wise additions are independent. If it is appropriate, the analyzer tries to, e.g. partition a DO loop to achieve higher parallelism. At this point, the analyzer consists of three major parts, Masterprogram, DO Loop Subprogram and Assignment Statement Subprogram. The details of each part are given subsequently.

9.1 Masterprogram (C. Cartegini)

The input information in the form of FORTRAN code is mapped into a suitable form to be used by the subprograms processing the tree height of arithmetic expressions within their own context. The purpose of the Master-program is initially to delimit each context and submit the related information in the form of parameters to the subprograms.

To do this, the stream of FORTRAN statements is partitioned into blocks characterized by the absence of transfer-control statements. These blocks are identified either by arithmetic assignment statements or DO-group statements. Non-executable FORTRAN statements are handled separately. The blocks as well as selective table-information are submitted to the subprograms.

The returned information in the form of unique or optional values is used as a weight associated to the nodes of a precedence graph identifying the original code and used in drawing the predictable behavior of the analyzed FORTRAN program as far as the rule of making no assumption in the actual value of input variables is not violated.

9.2 Assignment Statement Subprogram (J. Han)

This section describes the work of implementation of the tree height reduction algorithm discussed by Muraoka [3]. It describes two important properties, holes and spaces, within an arithmetic expression. By using distribution of multiplication over addition to fill these holes and spaces the tree height for an arithmetic expression can be reduced. The algorithm developed is based on some assumptions about a particular machine organization. Originally, this algorithm assumes, for simplicity, the operations of multiplication and addition have no difference in execution time between them. In the development of the implementation, this algorithm has been expanded to cover divisions and subtractions on the assumption that all kinds of operations take almost the same execution time.

So far the implementation of the tree height reduction algorithm has been programmed in PL/1 and tested. The first step of this implementation algorithm is to standardize an arithmetic expression. By this we mean to delete redundant parentheses for easy recognition of parallelism within it. Then we scan the standardized arithmetic expression to generate temporary results when one is found. Once the temporary result is to be generated, examination for the existence of holes and spaces is made to determine whether distribution of multiplication or division over additions and/or subtractions should be done. If one exists we need to rearrange the syntactic tree for the combination of the terms in their distributed form. If no distribution exists we simply build a syntatic tree for it. This algorithm uses a multipass-scanning mechanism. The number of passes required is not the tree height of an arithmetic expression, but depends on the depth of parenthesis nesting of an expression.

9.3 Do Loop Subprogram (S. Chen)

The sequential and iterated operations of DO or similar statements appear most frequently in many programming languages and statistics have shown that a lot of parallelism is lying within these operations specified by the usual application programmer. These can be done in parallel on a multiprocessor system and this results in decreasing the computing time.

Some types of parallelism existing in a restricted class of DO statements have been found and proposed by Y. Muraoka [3], based on the ease of compiling for future multiprocessor machines. Algorithms for detecting these types of parallelism have been sought and implemented in PL/1 programs.

The more general a nested DO loop is, the more complicated it becomes to discover parallelism, partially because this property primarily depends on the value of the index expression associated with each array variable, which might in turn depend on a value derived at execution time. Under certain assumptions, a general algorithm has been established and implemented in PL/1 programs to test the parallelism in nested loops, by using the algorithms for simple cases as building blocks. By this general algorithm, different kinds of parallel operations can be formed in terms of the number of processors in use. Some experiments are going to be performed on a set of randomly selected FORTRAN programs and the result will be compared with the original sequential operations of single processors. Furthermore, the effect of these assumptions upon the result will be evaluated.

Conclusion

In the next quarter the above three parts will be put together and experiments on real application programs will be done. Besides, refinement of the analyzer is also planned. Elaboration includes: (1) the analysis of a program structure, e.g. IF statements, (2) the measurement of a program, e.g. the frequency count of operators.

9.4 Tree-Height Reduction With Weighted Operands And Graph Schedules (P. Kraska)

Given an arbitrary arithmetic expression of scalar operands, there are algorithms (Muraoka [3], Baer [1]) which parse these expressions into trees such that maximum parallelism is exposed (i.e. tree-height is minimized). Thus, presumably, a computing system of parallel processors could evaluate the expression in minimal time. However, these algorithms do not account for the fact that the binary operators multiply, divide, and add (subtract) require different amounts of time for execution. During this quarter we have developed algorithms which minimize tree-height with time-weighted nodes by appropriate parsing of the expressions. Furthermore, the algorithms permit analysis to determine when distribution of multiplication and division across summation will reduce tree-height.

Tree-height minimization of arithmetic expressions of matrix operands, where the matrices are non-square, is a much more difficult problem since matrix multiply is not abelian and each matrix multiply node has a weight of $W_m = W_a \log_2 d_i$, where d_i is the conformable dimension of $(M_i \; M_{i+1})$, W_m and W_a are the time-weights of the scalar operators multiply and add, respectively. However, some progress has been made to the solution of this problem and it is conjectured that we can inductively prove that the tree-height of a product of n matrices may be minimized.

Tree-height reduction sometimes increases the number of nodes; thus any finite parallel processing system may be burdened with more work while evaluating the expressions. Evaluation of common subexpressions only once keeps the number of nodes to a minimum, but this introduces another problem; while algorithms exist which schedule m processors on a tree

(t. C. Hu [2]), there are currently no published algorithms which schedule m processors (m > 2) on a graph. However, we have had some success in this area. By analyzing an ordered connectivity matrix of a reduced directed graph it appears that cut sets of order m (i.e. m nodes which are mutually independent) may be extracted. It is encouraging to note that a recent paper by Ramamoorthy [4] uses this same technique. The problem of scheduling m-processors where weighted nodes comprise the graph is still unsolved.

9.5 Simulation Processor (E. Davis)

The purpose of this research is to study digital computer simulation of discrete time systems with the goal being the design of a machine for simulation processing. To achieve a significant factor of improvement in processing speeds a multiprocessor organization is being examined.

An interesting aspect of the design is that a range of the real time being simulated can be involved in the processing being done concurrently. That is, at an instant of computer time, more than one instant of simulated real time can be in process. This implies all the real time interactions plus those caused by the processing overlap of events at distinct times must be considered.

The problem then is to resolve all the processing interactions in a way that tends to maximize the processing that can be done concurrently.

Simulation languages have been studied. They reveal potential for concurrent processing due to many transactions flowing through a single program, corresponding to many executions of the program.

Resolution of the run time precedence requirements is being designed into a control unit. Output of the unit is tasks for the processors that make up the multiprocessor configuration.

9.6 Memory Hiera:chies (D. Gold)

Memory speed and memory cost are roughly inversely related. This indicates a significant potential savings for systems which have problem mixes requiring large amounts of memory.

Specifically, an ideal system (memory-cost-wise) will have only as much primary memory as is required by the basic operations (kernels) in its problem mix. The remainder of its memory being slower - and hence cheaper.

One of the problems in devising such a system is that of making the bulk-storage device look like a random-access device. These memories (for obvious cost considerations) are typically periodically addressable memories: disks, drums, recycling delay lines, etc. This has been accomplished for several large classes of problems by having a memory heirarchy consisting of primary memory, large periodically addressable memory, and relatively small bulk (random-access) memory. This bulk memory acts as a buffer to the other memories and early results indicate that it need be no larger than approximately one percent the storage capacity of the periodically addressable memory.

Another problem being investigated is that of precisely defining those classes of problems for which solutions have been found. Thus far, two- and three-dimensional mesh problems and most matrix operations have been found to be contained in these classes.

Included in these results is a method whereby any set of data objects (mesh points, partitions of matrices, etc.) may be successively randomly permuted between operations on them.

9.7 Microprogramming and File Processing (L. Hollaar)

During the last quarter, prototype 3 of the Burroughs D-machine was checked out and is now functioning correctly. However, since the machine lacks input/output facilities, the only programs run on the machine have been short diagnostic routines. Current plans are for this machine to act as an input/output controller for the new processor from Burroughs.

The card reader and the printer have been loaned to the PDP-11 project; assistance was provided for the installation and interfacing of these two devices.

Planning for the interface of the disk to the new processor has begun. As many disk functions as possible have been checked and the crystal clock used during writes have been received.

9.8 Text-Searching Project (W. Stellhorn)

A principal application planned for the D-machine is the investigation of algorithms for searching large volumes of textual material for specific information requested by a user. Such a system would differ from most existing information retrieval systems in that it would operate directly with original text rather than with keyword lists or abstracts. Of particular interest are techniques for rapid text searching and for assisting the user in eliminating material which satisfies his request but which is of no interest.

To this end, preliminary investigations are beginning in the general area of information retrieval. Also under study are potentially useful properties of English prose, such as word frequencies and the characteristics of intervals which typically separate occurrences of words of various classes.

A data base consisting of 67 technical articles in digital form for initial experimentation has been obtained from Lehigh University and is presently being reformatted.

9.9 D-Machine Micro-Programs (H. Yamada and W. Stellhorn)

D-machine micro-programs for interpreting the S-language have been completed and substantially debugged. No further development or testing of these codes is planned until after installation of the computer. These are described in a forthcoming M.S. thesis by Hirohide Yamada [5].

9.10 S-Language Assembler (E. Polley)

In order to write programs for the D-machine it is necessary to have the instructions in the form required for Yamada's interpreter [5]. To prepare instructions for the interpreter they will first be written in an assembly language, created for the project, then translated into the object code which can be accepted by the interpreter. This translation process will be done by an assembler which is currently being finished.

The assembly language has two types of instructions: word and string. Word instructions are similar to those found on most machines, are of a three address format and include operations such as addition, subtraction, shifting and branching. The arithmetic instructions all have two forms one of which checks for overflow.

The other type of instructions, the string instructions, do the character manipulation which is the heart of this project. These instructions search for characters, compare strings and move strings in core.

The assembler presently can handle the word instructions only. It is a 360 assembly language program and will punch its object code into cards which will be read by the reader attached to the D-machine and loaded for interpreting.

9.11 Debugging (M. Kaplan)

Since September of 1970, we have been debugging a set of statements to be added to the WATFIV compiler to aid users in debugging logical errors in their programs. These statements include three kinds of traces - a line trace to trace the flow of execution, a variable trace to trace the flow of specific variables during the execution of a program, and a subprogram trace which will trace the invocation of function and subroutine subprograms and also print out the values of their parameters upon entering and leaving the routine. These traces will be similar to those provided in the FORTRAN-G compilers' debugging package but will be more flexible. For example, it will be possible to control their use at execution time. The other statement in the package is, as far as we know, a new idea. The rationale for this statement is that people write and debug programs in blocks. What this statement attempts to do is to provide an easy way to debug these blocks of code. The statement is of the form:

```
DEBUG < statement # > INITIAL < list of assign. statements > AT
STATEMENT < statement # > < list of logical exp. > AT STATEMENT
< statement # > < list of logical exp. > GO TO < statement # >
```

for example:

```
DEBUG 10 INITIAL X = 10.4, Y(1) = 12*Z**2 AT STATEMENT 20
X.EQ.20.5, I.LT.12 GO TO 50
```

The effect of this statement is to execute the block of code from the debug statement to statement number 10 with the initial conditions specified by the INITIAL clause. When it reaches statement number 20, it tests if X.EQ.20.0

and if I.LT.12 and prints out appropriate messages. The GO TO 50 clause causes the DEBUG package to ckeck that when the program leaves the DEBUG block it goes to statement 50.

A statement similar to the PL-I ON statement is also under consideration.

For the past three semesters, CS 109 students have been using, modifying and writing about a partial implementation of the DEBUG statement written as a PL-I prepass to WATFIV. Tests with these students, although not conclusive, show a tendency for them to debug the logic of their programs faster if they use the statement than if not. Further tests will be conducted with larger samples of CS 101 students this summer.

At the present the implementation into the WATFIV compiler of all the statements described is proceeding with completion hoped for by June.

LIST OF REFERENCES

[1] Baer, J. E., "Graph Models of Computations in Computer Systems," Ph.D. Dissertation, University of California, Los Angeles, Report No. 68-46 (October, 1968).

[2] Hu, T. C., "Parallel Sequencing and Assembly Line Problems," Operations Research, 9 (November-December, 1961), pp. 841-848.

[3] Muraoka, Y., "Parallelism Exposure and Exploitation in Programs," Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Report No. 424 (February, 1971).

[4] Ramamoorthy, C. V., and Chang, L. C., "System Segmentation for the Parallel Diagnosis of Computers," IEEE Transactions on Computers, C-20 (March, 1971), pp. 261-270.

[5] Yamada, H., "Emulation of Disc File Processor," M.S. Dissertation, University of Illinois at Urbana-Champaign, Report No. 436 (June, 1971).

10. COMPUTER SYSTEMS ANALYSIS

(Supported in part by the National Science Foundation under Grant No.
US NSF GJ 28289.)

During this report period the National Science Foundation initiated Grant No. US NSF GJ 28289 for research entitled "Computer Systems Analysis."

The goal of this research is the development of analytical tools for system modeling and analysis of real time computer networks. The particular network being investigated is that of a geographically distributed network of computers. A queueing theory model for this computing system based on the essential characteristics of the network, and priority assignment rules for efficient job processing at each of the computing centers of the network is being investigated.

10.1 Computer Network Modeling

We have begun this research in Computer Systems Analysis by formulating a mathematical model for a geographically distributed network of computers. Our current effort in this area is aimed at developing a queueing theory model for a multiserver system with a finite length queue. To date we have considered nonpriority queue disciplines and we are currently investigating nonpreemptive priority disciplines. Our future endeavor will include modeling preemptive priority and dynamic priority queues.

10.2 Center Throughput Analysis

We have begun research in this area by focusing our attention on Illinet, a geographically distributed computing center which provides online express, teletype timesharing, and remote batch entry services to a network of users at the University of Illinois. Based on the essential characteristics of the various computers within the center, we shall employ our queueing theory model to facilitate studying the effects of priority assignment and job dispatching.

Further effort in this area will be deferred until the queueing theory model is defined.

(E. K. Bowdon)

11. ILLIAC IV

(This work was supported in part by the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.)

REPORT SUMMARY

Automation Technology, Inc. reports that PE tests and Card tests proceeded well during the quarter. Diagnostic efforts will be concentrated on "diagnosing" conversion problems through a major portion of the next quarter.

During this quarter the ILLIAC IV Project has received a Burroughs B6500/B6700 Computer at the University of Illinois. This machine has been obtained primarily to support the programmers developing the ILLIAC IV operating system. It is also available to applications programmers and to prospective users of the ILLIAC IV computer.

The software effort was directed toward integrating the Operating System with the ILLIAC IV simulator on the B6500, converting the Cockroach-to-Glypnir translator to the B6500, supporting the assembler (ASK) on the B6500, and simulating ILLIAC IV special functions and algorithms written in ASK for a mathematical subroutine library. The Software Reference Manual is nearly complete.

Several major software efforts for the PDP-11 system were continued or completed. Hardware delivery began on the various segments of the ARPA network terminal system. Negotiations were completed and agreement reached with Digital Equipment Corporation on a co-operative research arrangement for joint development of the ARPA network port facility.

Application efforts in Numerical Analysis continue in solving partial differential equations, matrix inversion and linear algebraic equations, eigenvalues and eigenvectors, estimation and filtering, and identification of nonlinear differential equations.

Three seminars were offered during the quarter.

Project expenditures and commitments through March, 1971:

Burroughs Corporation	$26,723,000.00
University of Illinois	6,814,273.83

HARDWARE

11.1 Off-Line Diagnostics

11.1.1 PE Test

The analysis of the PE logic and the development of tests to be used to test the PE off-line through use of the PE Exerciser (PEX) is progressing well. The Path Tests and Combinational Logic Tests are completed and in use. The analysis necessary for development of tests of the Control Logic is nearing completion and some test sequences have been defined. This area is expected to be complete by late summer.

11.1.2 Card Test

The development of tests to isolate stuck-type faults in active devices is on schedule except as noted below. All tests for PE Cards are complete, have been verified, and are in use. Of the CU Cards, the Type 1 boards are scheduled to be completed by mid-summer. The type 1 boards have been divided into two classes, 31 Priority Boards and 40 Non-Priority Boards. Twenty-eight Priority Boards and 21 Non-Priority Boards have their tests generated; however, these tests are not verified, nor are they in use, since there is no CU Card Tester operational.

11.2 Program Conversion

Conversion of the program used to develop the diagnostics mentioned above was begun during the quarter. Because of the importance of conversion to be able to carry on useful test generation, some other scheduled work is being postponed. The level of effort on the PE Control Logic Tests is temporarily reduced to analysis only and no board tests are being generated. The majority of diagnostic efforts will be concentrated on "diagnosing" conversion problems through a major portion of the next quarter.

11.3 ILLIAC IV Maintenance

ATI maintenance engineers are actively participating in the on-line debugging of ILLIAC IV. Among the areas of active participation have been: PE Board Test, PEM/MLU Test, PE Test, CU Test and I/O Test. In addition, ATI has identified the long lead spare parts and estimated the required number of each. Under authorization of the University of Illinois, ATI is obtaining competitive prices, where possible.

11.4 Financial

At this point in the contract, ATI is approximately ten percent (10%) below the budgeted cost estimates.

SOFTWARE

11.5 Operating System

11.5.1 Operating System I

The operating system is at present being integrated with the ILLIAC IV simulator (SSK) on the B6500, and has reached the stage where simple ILLIAC IV programs have been run under the control of the ILLIAC IV operating system. Instrumentation and error diagnostics are now being placed in some of the modules to allow the performance of the operating system to be measured in detail. An Operating System Maintenance Manual has been written.

11.6 Compilers and Translators

11.6.1 Glypnir

Glypnir, Version II, has been undergoing further consolidation. I/O for ILLIAC IV (as opposed to Simulator I/O) and Macro Facilities in the form of a pre-processor have been provided. The Glypnir compiler has also undergone a basic measurement investigation with a view to speeding it up. The possibility of providing a facility for separately compiling subroutines has been and is still being studied. A Glypnir Compiler Maintenance Manual has been written.

11.6.2 Cockroach

During the quarter the Cockroach-to-Glypnir translator was completed for a subset of the specified language on the B6500 and converted to the B6500. Subroutines and functions will be completed in the beginning of the second quarter. User documentation of the available features was also provided. The addition of an hourly employee to the staff hastened the completion and provided for more extensive debugging.

Cockroach is now available to users, and a certain limited amount of user support also is available.

11.7 Assembler

The Assembler (ASK) is now supported on the B6500. It compiles at about 1200 cards per minute. A plan for increasing its compiling speed to 2500 cards per minute has been formulated and is being implemented.

11.8 Interactive Communications and Graphics

11.8.1 Interactive Communications

During the reporting period several major software efforts for the PDP-11 system were continued or completed.

1. A high-level language compiler, PEESPOL, was completed. Version II is running on the B6500.
2. Version I of the ARPA Network Terminal System, ANTS, was completed. Basic portions of the system were checked out and actual runs made on the PDP-11.
3. Design of the link-up between Paoli and the University of Illinois, via the ARPA network, using two ANTS systems was completed and initiated.
4. Final specifications were agreed upon by Burroughs for the construction of an interface between the B6500/B6700 system and the ARPA network IMP.

During this reporting period, hardware delivery began on the various segments of the ARPA network terminal system. The basic PDP-11 processor, the 16K words of memory, the real-time block, the high-speep paper tape, ASR35 operator's teletype, four 2400 Baud line interfaces with adapters, one dataset control and interface for a Bell 103A dataset, plus two general-purpose interfaces to be used to connect the Gould Electrostatic Plotter to the system and to connect the Computek storage scope system were received. The remainder of the items ordered should be delivered within the next reporting period.

Two interfaces between the ARPA network IMP and the PDP-11 were received. Both interfaces were debugged, checked out, and installed in the University of Illinois PDP-11 and the Paoli PDP-11. Success in getting on the network now hinges on the completion of system software development.

Negotiations were completed and agreement reached with Digital Equipment Corporation on a cooperative research arrangement for joint development of the ARPA network prot facility.

11.8.2 Graphics

No work was done in the graphics area during this reporting period because of total lack of personnel. An effort was begun, toward the end of the period, to acquire a full-time professional in graphics who is expected to join the Center during the next period.

11.9 Mathematical Subroutines Special Function Library

This quarter's work has continued on the ILLIAC IV Special Functions and Algorithm subroutine library. The following set of routines written in ASK have been successfully simulated by the current version of the simulator and the results are good for 15 significant digits:

64-bit mode:
- sine and cosine
- tangent and cotangent
- natural logarithm
- square root
- arctangent (1 argument)
- arctangent (2 arguments)
- hyperbolic sine and cosine
- hyperbolic tangent
- gamma range(0,1)
- gamma range (0,∞)

32-bit mode:
- sine and cosine
- tangent and cotangent

- exponential
- natural logarithm
- square root

Algorithmic library:

- addition/subtraction of matrices stored straight or skewed
- multiplication of matrices stored straight
- transpose of a matrix stored straight

At present work is being completed for:

64-bit mode:
- gamma (X) $0 < X \leq 57$
  $57 < X < \infty$ and
  $X < 0$
- error function: erf (X) and
  LN gamma (X).

For matrix operations the set is being expanded to rectangular matrices stored straight and skewed. The same set will also be made available in Glypnir. These routines will continue to be refined. Maintenance, distribution and programming assistance in using the mathematical routines is available.

11.10 B5500/B6500 Operations

During this quarter the ILLIAC IV Project has taken delivery of a Burroughs B6500 computer at the University of Illinois. This machine has been rented primarily to support the programmers developing the ILLIAC IV operating system. It is also available to applications programmers from the Center for Advanced Computation and future users of ILLIAC IV. The machine was installed in the basement of the Coordinated Science Laboratory and became operational around March 1.

11.11 Software Documentation

The Software Reference Manual is now at 900 pages, largely completed. Its one deficiency is an Assembler users manual, which is at present about half completed.

11.12 Numerical Analysis

11.12.1 Partial Differential Equations

11.12.1.1 Numerical Solution of Problems in Hydrodynamics

The results of numerical experiments with the Brailovskaya, Dufort-Frankel, Cheng-Allen, Crank-Nicholson and Lax-Wendroff finite difference schemes on the Burgess equation have been presented in a formal report which is now in the process of being printed [1].

A code is being written and debugged for the two dimensional subsonic and transonic flow around a circular cylinder, as the next step in exploiting fully the techniques of parallel processing in the numerical computations of three-dimensional fluid and gas flow problems. In a polar co-ordinate system the governing equations in non-dimensional form are:

Continuity $$\frac{Dp}{Dt} + \rho\nabla \cdot \vec{V} = 0;$$

Momentum $$\rho \frac{D\vec{V}}{Dt} + \nabla p = \sqrt{\gamma}\, \frac{M_\infty}{Re} \left(4/3\ \nabla\nabla \cdot \vec{V} - \nabla \times \nabla \times \vec{V}\right);$$

Energy $$\rho \frac{DS}{Dt} = \sqrt{\gamma}\ (\gamma - 1)\ \frac{M_\infty}{Re}\, \Phi + \gamma\, \frac{\sqrt{\gamma}\ M_\infty}{Re \cdot Pr}\, \nabla^2\ (p/\rho)$$

$\vec{V}$ = velocity, p = pressure, S = entropy, γ = ratio of specific heats, M_∞ = free stream Mach number, Φ = dissipations terms, Re = Reynolds number, Pr = Prandtl number.

The computational region comprising the flow field is divided into a suitable number of mesh points, which is a function of computer storage capacity, and the dependent variables, namely the pressure p, radial and tangential components of the velocity, i.e. u, v, and temperature, T, are calculated at each mesh point. The equation of state for a perfect gas is assumed. Initially the gas is assumed to be flowing with a uniform velocity and the physical properties are assumed uniform at each mesh point.

To gain flexibility and save time on runs while the program is being debugged, the viscous terms have been suppressed in the calculation, so that the flow is treated as being inviscid. At time t = 0, the condition of zero radial velocity is imposed on the cylinder boundary, a suitable time step is selected, and the finite difference representation of the governing equations is used to calculate the values of the dependent variables at succeeding times. The finite difference scheme, chosen for these experiments is the two-step, second order accurate, Richtmyer variation of the Lax-Wendroff scheme [2], which has been previously tested with the Burgess equation [1].

11.12.1.2 Algorithm Development

During this period several numerical methods have been investigated in cooperation with AMOCO Oil Company to solve the elliptic type partial differential equations. The semi-iterative block Jacobi method and ADI method were adopted and both were implemented in Glypnir. The first method worked correctly and was made into a general type subroutine. An iterative step takes about 600 microseconds on ILLIAC IV in the case of 21 x 21 mesh points. The second method is still being implemented. The storage scheme for this method is more complicated than the first method.

11.12.2 Matrix Inversion and Solution of Linear Algebraic Equations

During this quarter, a program to solve linear equations and perform matrix inversion has been written in ASK, and is presently being debugged. It uses the Gaussian Elimination method and is designed to handle full matrices up to 630 x 630 for Gaussian elimination and up to 475 x 475 for inversion. These large matrices are not core containable and are therefore broken down into blocks of 64 columns, each block being processed independently of the others as far as possible.

The algorithm works in two stages, first the matrix A is decomposed into the product of lower and upper triangular matrices, L U.

The second stage uses this decomposition to operate on a right hand side to solve Ax = b, on a unit matrix to form A^{-1} or on an arbitrary matrix B to form $A^{-1}B$. To reduce the round-off error, 'partial pivoting' is employed (see e.g. Wilkinson, J. H., The Algebraic Eigenvalue Problem), also the inner products of vectors are accumulated using double precision arithmetic.

Work is just beginning on the Conjugate Direction method which will be implemented in Glypnir. This should prove an efficient algorithm for solving Ax = b when the matrix A is not given explicitly but a subroutine which computes Ax given the vector x is available.

11.12.3 Eigenvalues

Jacobi's Method for finding eigenvalues and eigenvectors of real symmetric matrices (including complex Hermitian matrices: Let $A = B + iC$ be a complex Hermitian matrix where B is real symmetric ($B = B^t$) and C skew-symmetric ($C = -C^t$), then the real symmetric 2n x 2n matrix

$$A^1 = \left[\begin{array}{c:c} B & -C \\ \hdashline C & B \end{array}\right]$$

can be an input to the Jacobi algorithm) has been coded in ASK and satisfactorily tested on the B5500 ILLIAC IV simulator.

Eberlein's Method (Jacobi-like algorithm) for normalizing real non-symmetric matrices has also been satisfactorily tested.

The results of both algorithms agree with the literature up to 14 significant digits. An error of order O (10^{-15}) is due to conversion routines from internal machine representation of floating points to their external decimal representation.

To have a means of comparison, two well established algorithms by H. Rutishauser [3] and P. J. Eberlein [4] have been combined in one B6700 ALGOL program. It is in a running condition and exists under the file name (completed version) WINFRIED/BERNHARD/PROC. This file is a separately compiled procedure. The name of the procedure is JACEIG (N,A,TS,D,ROT,TMX).

In using this procedure the user has an option

1) He may write his own main-program for I/O and procedure call, but he then has to go through the B6700 BINDER for interrelation between his file and the above file or,

2) He can use the file BERNHARD/JACOBI/EIGEN which does the I/O and calculation for him.

For case (2) the following control cards and input-data are required:

```
δ EXECUTE BERNHARD/JACOBI/EIGEN
δ BCL EBOR
<integer 1>, <integer 2>,
matrix-elements in real.
δ END
```

where δ is an invalid character like multipunch 1,2,3

<integer 1> = order of matrix

<integer 2> = TMX with

TMX = 0 no eigenvectors are calculated and printed

TMX > 0 the right eigenvectors are calculated and printed

TMX < 0 the left eigenvectors are calculated and printed.

The matrix for which the eigenproblem is to be solved can be either real symmetric or real non-symmetric. If real symmetric then only the eigenvalues are printed in decreasing order. If the eigenvectors are to be printed (TMX ≠ 0), then this will leave the same order as the eigenvalues, i.e., to λ_j corresponds TS_j in the order of their appearance.

If real non-symmetric the total eigenvalue matrix is printed. If the eigenvalues are complex then the a_{ii} occupy the diagonal while the imaginary part occupies the off diagonal portions.

If $a_{ii} \pm i\ a_{ij}$ is the eigenvalue, the corresponding eigenvector $t_i \pm i_{(t_j)}$ appears in column (row) i and column (row) j.

11.12.3.1 Eigenvalues and Eigenvectors of Symmetric Tridiagonal Matrices

11.12.3.1.1 QR Algorithm with Origin Shifts

The QR algorithm for finding eigenvalues of symmetric tridiagonal matrices has been written in XALGOL, debugged and tested for different-size matrices. A short description of the algorithm follows.

A is a symmetric tridiagonal matrix with diagonal elements C_i and subdiagonal elements b_i. Any symmetric tridiagonal matrix A_t, $(t = 1, 2, \ldots)$, may be expressed as

$$A_t = Q_t R_t \tag{1}$$

where Q_t is orthogonal and R_t is upper triangular of the form

$$\begin{bmatrix} p_1 & q_1 & r_1 & & & 0 \\ & p_2 & q_2 & r_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & r_{n-2} \\ & & & & p_{n-1} & q_{n-1} \\ 0 & & & & & p_n \end{bmatrix}$$

Let $A_{t+1} = Q_t^T A_t Q_t$ then $A_{t+1} = R_t Q_t$. (2)

For large t, A_t approaches a diagonal matrix, Diag $[\lambda_i]$, where λ_i's are the eigenvalues of the matrix, A.

To accelerate convergence of this algorithm, we let

$$A_t' = A_t - K_t I$$

where K_t is the eigenvalue of the 2 x 2 matrix

$$\begin{bmatrix} c_{n-1}^{(t)} & b_{n-1}^{(t)} \\ b_{n-1}^{(t)} & c_n^{(t)} \end{bmatrix}$$

which is closer to $C_n^{(t)}$. Then we decompose $A_t' = Q_t R_t$ and find

$$A_{t+1} = R_t Q_t + K_t I.$$

By repeating these two steps, the eigenvalues of the matrix, A, are immediately available.

A document describing the algorithm with flow chart is being written.

11.12.3.1.2 Inverse Iteration to Find Eigenvectors

A program to calculate the eigenvectors of symmetric tridiagonal matrices with given eigenvalues has been written in XALGOL, debugged and tested. The algorithm is as follows:

A is a symmetric tridiagonal matrix with c_i as its diagonal elements and b_i as its subdiagonal elements. Its eigenvalues are given rather accurately and arranged in descending order.

Let vector $\vec{V}$ be taken as the initial vector. We solve

$$(A - \lambda I)\ \vec{X} = \vec{V} \tag{1}$$

followed by

$$(A - \lambda I)\ \vec{Y} = \vec{X} \tag{2}$$

then $\vec{Y}$ is the eigenvector corresponding to eigenvalue λ.

To make the algorithm more effective, we let

$$(A - \lambda I) = LU \tag{3}$$

where L is unit lower triangular and U is upper triangular, set $\vec{V} = L\vec{e}$ where $\vec{e}^t = (1, 1, \ldots, 1)$, then

$$U\ \vec{X} = \vec{e}. \tag{4}$$

Hence $\vec{X}$ can be determined by back-substitution. The matrix U is determined by Gaussian elimination with partial pivoting, i.e. eliminate the variables in their natural order, but select the row having the maximal coefficient of X_i as the pivotal row in the i-th stage. Similarly, we can find the vector $\vec{Y}$.

A document describing the algorithm with flow chart and numerical examples is being written.

11.12.4 Polynomial Root Finder

Actual coding of a root finder has not been done in ASK. However ALGOL codings have been debugged on the B5500. Presently changes are being made in these programs to run on the B6500.

In general, the algorithm has been outlined to be: (1) evaluate polynomial at various intervals to determine whether a change of sign takes place, (2) shorten these intervals as much as possible, (3) proceed on new intervals with Newton-Rapheson iterations. Also an ALGOL program using Sturm sequences to evaluate intervals is presently being debugged on the B6500.

11.12.5 Identification of Non-linear Differential Equations

Much is known about the numerical solutions of differential equations, however, almost nothing is known about the reverse problem, i.e., given some observed function of time, x(t), can we find a differential equation of which x(t) is a solution?

Such problems exist in economics, chemistry, medicine and many other fields, the equations involved often being nonlinear. In the past solution has been restricted to linear problems or problems involving the estimation of only a few parameters. This has been caused not so much by lack of solution techniques, but by inadequate computing power. ILLIAC IV can alleviate this problem through its high computing speed and the fact that the problem can utilize the array processor efficiently.

Since many different differential equations may have the same solution, it is necessary to restrict the problem to time invariant equations of the form:

$$F(x, x', x'', \ldots, x^{(m)}, f_1, \ldots, f_n) = 0 \tag{1}$$

where F is assumed to be known,

$$x^{(j)} \equiv \frac{d^t}{dt^j} x(t) \text{ and}$$

$$f_i \equiv f_i(x^{(j_i)}) \text{ are unknown functions}$$

of x or one of its derivatives. The problem is to develop an algorithm to determine the f_i.

$$\text{E.g.} \qquad x'' + f(x) = 0.$$

If x were given as $x(t) = \sin t$, we could derive $f(x) = x$. Let $\hat{x}(t)$ be the observed function, defined on the interval $0 \leq t \leq T$.

The solution to (1) depends on the functions f_i and the initial conditions of x and its first n - 1 derivatives at t = 0.

I.e., $x = x\,(x_0, x_0', x_0'', \ldots, x_0^{(m-1)}, f_1, \ldots, f_n)$

or more briefly $x = x(\underset{\sim}{x}_0, \underset{\sim}{f})$.

We define the error functional

$$E\,(\underset{\sim}{x}_0, \underset{\sim}{f}) = \|\, \hat{x} - x\,(\underset{\sim}{x}_0, \underset{\sim}{f})\, \|^2$$

where

$$\|\, y \,\|^2 = \int_0^T w(t)\,[y(t)]^2\,dt$$

for some weighting function, w. E is a measure of the difference between observed function and the computed one for some $\underset{\sim}{x}_0$ and $\underset{\sim}{f}$.

The problem thus reduces to finding $\underset{\sim}{x}_0$ and $\underset{\sim}{f}$ such that $E\,(\underset{\sim}{x}_0, \underset{\sim}{f})$ is minimized.

For the numerical solution, $\hat{x}(t)$ is assumed to be observed at discrete, equal intervals and the integrals replaced by summations. It is also assumed that these observations are contaminated by random "noise" so that minimizing E constitutes a least squares solution to the problem.

The functions f_i are approximated over the interval on which they are to be identified, [a,b], by a linear combination of fixed orthonormal functions

$$f(x) \approx \sum_{j=1}^{n} q_j\,\Phi_j\,(x)$$

where $< \Phi_i, \Phi_j >_1 = \delta_{ij}$, the Kroneuker delta function, the inner product $< g, h >_1$ being given by $< g, h >_1 = \int_a^b g\,(x)\,h\,(x)\,dx$. In this implementation the functions, Φ_j, will be splines.

The Solution Algorithm

For simplicity, we consider the problem

$$\frac{dx}{dt} = f(x)$$

Let $f(x) = \sum_{i=1}^{n} q_i \Phi_i (x)$ and $x(0) = x_0 = q_0$. Define the solution of

$$x' = f(x)$$
$$x(0) = x_0$$

to be $x_f(t)$. The dependence on x_0 being implicit.

Then $E(f) = E(q) = ||\hat{x} - x||^2$

$$= \int_0^T [\hat{x}(t) - x_f(t)]^2 \, dt.$$

When $E(q)$ is minimized, $\nabla q E = 0$

where $(\nabla q E)_i = \frac{\partial}{\partial q_i} E$. $i = 0, 1, \ldots, n$.

A vector q such that $\nabla q E = 0$ may be found iteratively using the Conjugate Gradient Method [1] :

$$q_{n+1} = q_n + c_n p_n$$

where: $p_0 = - (\nabla q E)^{(0)}$,

$$p_{n+1} = - (\nabla q E)^{(n+1)} + b_n p_n \, , \tag{2}$$

$$(\nabla q E)^{(n+1)} = \nabla q E(q_{n+1}) \, ,$$

$$b_n = - \frac{< (\nabla q E)^{(n+1)}, (\nabla^2 q E)^{(n+1)} p_n >_2}{< p_n \, , (\nabla^2 q E)^{(n+1)} p_n >_2} \, ,$$

$$(\nabla^2 q E)^{(n+1)} = \nabla^2 q E (q_{n+1})$$

and $\nabla^2 q E$ is the matrix of second derivatives:

$$(\nabla^2 q E)_{ij} = \frac{\partial^2}{\partial q_i \, \partial q_j} E \, .$$

The superscripts indicate the iteration count. The inner product $< a,b >_2$ being defined by

$$< a,b >_2 = \sum_{i=0}^{n} a_i \, b_i \; .$$

At each iteration, c_n is chosen so as to minimize $E(g_n + cp_n)$ over all c.

The calculation of $\frac{\partial}{\partial q_i}$ E .

$$\frac{\partial}{\partial q_i} E = \frac{\partial}{\partial q_i} \int_0^T w(t) \, [\hat{x}(t) - x_f(t)]^2 \, dt$$

$$= -2 \int_0^T w(t) \, [\hat{x}(t) - x_f(t)] \; . \; \frac{\partial x_f}{\partial q_i} (t) \quad dt \qquad (3)$$

if we abbreviate $\frac{\partial x_f}{\partial q_i}$ by $\delta\, x_i$, then $\delta\, x_i$ satisfies the so-called "sensitivity equations"

$$\frac{d}{dt} \; \delta\, x_0 = \frac{\partial f}{\partial x} (x_f) \; . \; \delta\, x_0, \; \delta\, x_0 \; (0) = 1$$

and $$\frac{d}{dt} \; \delta\, x_i = \frac{\partial f}{\partial x} (x_f) \; . \; \delta\, x_i + \Phi_i \; (x_f), \; \delta\, x_i = 0, \; i = 1, 2, \ldots, n \, .$$

representing the sensitivity of the equation x' = f(x) to changes in x_0 and f respectively.

If we define $\Phi_0 \; (x) \equiv 0$, these may be combined to:

$$\frac{d}{dt} \; \delta\, x_i = \frac{\partial f}{\partial x} (x_f) \; . \; \delta\, x_i + \Phi_i \; (x_f) \quad i = 0, 1, \ldots, n \; . \qquad (4)$$

To determine ∇^2 q E we note that,

$$\frac{\partial^2 E}{\partial q_i \, \partial q_j} = \frac{\partial^2}{\partial q_i \, \partial q_j} \int_0^T w(t) \, [\hat{n}(t) - n_f \, (t)]^2 \, dt$$

$$= - 2 \int_0^T w(t) \; \frac{\partial}{\partial q_i} \; x_f(t) \; \frac{\partial}{\partial q_j} x_f(t) \quad dt$$

$$- 2 \int_0^T w(t) \, [\hat{n}(t) - x_f(t)] \times \frac{\partial^2}{\partial q_i \, \partial q_j} (x_f) \; (t) \; dt$$

$$= 2 \int_0^T w(t)\ \delta\, n_i\ \delta\, n_j\ dt$$

$$- 2 \int_0^T w(t)\ [\hat{n}(t) - x_f(t)]\ \delta\, n_{ij}^2\ dt \qquad (5)$$

where we define $\delta\, n_{ij}^2 \equiv \frac{\partial^2}{\partial q_i\ \partial q_j}\ (x_f)\ (t)$.

By differentiating (t) with respect to q_j, $\delta^2 n_{ij}$ may be shown to satisfy the equation:

$$\frac{d}{dt}\,\delta^2 n_{ij} = \frac{\partial^2 f}{\partial n^2}\ \delta\, n_i\ \delta\, n_j$$

$$+ \frac{\partial f}{\partial n}\ \delta^2 n_{ij}$$

$$+ \frac{\partial}{\partial n}\ \Phi_i\ \delta\, n_j + \frac{\partial}{\partial n}\ \Phi_j\ \delta_{ni} \qquad (6)$$

Since we require only ∇^2 q E explicitly, we need calculate only the components $\Psi_i \equiv \sum_{j=0}^{n}\ \delta^2 n_{ij}\ p_j$.

Multiplying (6) by p_j and summing over j gives:

$$\frac{d}{dt}\ \Psi_i = \frac{\partial^2 f}{\partial n^2}\ \delta\, n_i\ (\ \Sigma\ \ \delta\, n_j\ p_j)$$

$$+ \frac{\partial f}{\partial n}\ \Psi_i$$

$$+ \frac{\partial}{\partial n}\ \Phi_i\ \ (\ \Sigma\ \ \delta\, n_j\ p_j)$$

$$+ (\frac{\partial}{\partial n}\ \Sigma\ \ \Phi_j\ p_j)\ \delta\, n_i \qquad (7)$$

where all summations are from j = 0 to n. This reduces the number of differential equations to be solved from $(n + 1)^2$ to n + 1 .

Using (5) we obtain

$$[\nabla^2 q \; E \; p]_i = 2 \int_0^T w(t) \;\; \delta \, n_i \; x \; (\sum \delta \, n_j \, p_j) \; dt$$

$$- 2 \int_0^T w(t) \; [\hat{n} \; (t) - x_f(t)] \; \Psi_i(t) \; dt \; .$$

Preliminary experiments indicate that this algorithm is rapidly convergent, only 5-10 iterations being required when a good estimate of f is available.

Implementation on ILLIAC IV

A specialized compiler is needed to perform the following tasks:

1. Accept the form of the equation to be identified.
2. By using symbolic differential techniques, form the sensitivity equations.

11.12.6 Estimation and Filtering

11.12.6.1 Non-linear Least Squares Problem

Work is proceeding on the parameter estimation of non-linear least squares problems. The numerical algorithms to be investigated are:

1. modified Gauss method
2. gradient method
3. variable metric method
4. factorization method

The implementation of algorithm (1) has been successfully completed and the ASK code for (2) is proceeding for which more accurate results are expected. Work has also begun on the variable metric method.

11.12.6.2 Numerical Solution of the Non-linear Matrix Riccati Equation

An eigenvector solution of the matrix Riccati equation relating to optimal control theory has been studied during this quarter. For the cases in which the matrix of coefficients resulting in linear dynamic systems can be transformed to a diagonal matrix, there is a straight forward method for constructing a similarity transformation whether or not the eigenvalues are distinct.

11.13 Linear Programming

This quarter has seen the switch to the B6500 computer installation. Regrettably, the transition involved a degradation of the B5500 assembler-simulator support for a prolonged period, without comparable facilities being available elsewhere. This seven week stretch was used to bring LP-system documentation into usable shape.

The simulation that could be performed was directed wholly toward the INVERT package. There are indications that simulation will not permit sufficient code "execution" to debug the inversion routines, but work is continuing.

B5500 SETUP routines have been run on the new computer, but require modifications and extensive updating to reflect the increasing load of SETUP that the ILLIAC IV will perform. Experience with our B6500 indicates that only the most minimal processing should remain on this easily overtaxed facility. The future will see consideration being given to transferring more functions, including those--such as sorting--which are ungainly and difficult to code for the ILLIAC.

The problem of removing the solution and associated information is now being resolved. Initially these SETDOWN procedures will be executed as a separate program as it is estimated that abnormal terminations will be frequent. Again, as with SETUP, the majority of the work--such as rescaling and transforming data to B6500 representation--will be executed on ILLIAC IV.

Documentation was reviewed, rewritten, and reorganized with the intent of improving the clarity of the system's interconnections and interpresumptions. Not surprisingly, discrepancies were found. Further, refinement will be necessary, but will be deferred until after the code has been run on the prototype ILLIAC.

11.14 Long Codes

An analysis of the noise vectors used for previous studies disclosed an alarming degree of correlation between the supposedly independent random vectors. Consequently, an improved pseudo-random number generator of proven period and potency was substituted for the previously used ad hoc generator. All identification algorithms tested were more effective when used with the less correlated noise vectors.

A Kalman filter algorithm is being implemented to allow testing of the parameters obtained from the identification algorithms without knowledge of the true values of the parameters. If the innovation sequence of a Kalman filter using the estimated parameters as a model and using the observed values as input is a white noise sequence, identification can be considered successful.

The most recently developed correlation approach for identification showed considerable promise before its use was suspended for recoding to run on the B6500. The equations for this scheme, like most other relations that occur in identification processes, include both the covariance matrix of the plant noise and the covariance matrix of the observation noise, which are usually unknown. Manipulation of the correlation algorithm expressions indicates that it will be possible to obtain not only estimates of the parameters of the system, but also estimates of the two covariance matrices. It may be necessary to require the correlation matrix of the observation noise to be in the form $\sigma^2 I$ (I = Identity matrix), but this restriction can probably be relaxed.

11.15 Signal Processing

A Glypnir subroutine has been written which computes the fast Fourier transform of any real vector of length N, where N is a power of two and $N \geq 64$.

A previously written Glypnir autocorrelation program has been put into the form of a subroutine.

Programs are being written for use on the B6500 computer which will be used to thoroughly test previously written ALGOL procedures and Glypnir subroutines.

Work is continuing with Amoco Production Company on implementing some signal processing and partial differential equations algorithms on the ILLIAC IV.

11.16 Education

11.16.1 ILLIAC IV Seminars

In addition to the graduate course on ILLIAC IV, three seminars were offered this past quarter: A one-day seminar was given on January 5, 1971. The outline follows:

Seminar on ILLIAC IV

Background	9:00 - 9:30 am
Buffer, Pipeline and Multiprocessor	
Hardware Structure	9:30 - 12:00 n
A General Description of Array A Sample Problem A More Detailed Description of Array A General Description of I/O System	
Test/Repair Equipment and Diagnostics	1:30 - 2:00 pm
Physical Characteristics	
Software	2:00 - 4:30 pm
Programming Languages -- ASK, GLYPNIR, FORTRAN Operating System Utilities	
Applications	4:30 - 5:00 pm

On January 11-15, a one-week seminar was given to Pan American Petroleum. On March 15-19, another one-week seminar was given to employees at Ames Research Center, Moffett Field, California, where ILLIAC IV is to reside. Additionally a one and one-half hour overview on ILLIAC IV was presented to about 300 Ames employees. The outline for the one-week seminar follows.

A Seminar on the ILLIAC IV System

I. Background -- Conventional and Unconventional Organizations

A. Why ILLIAC IV?
1. Conventional Organization
B. How to speed up the Operation-Design Philosophies
1. Overlap
a. Buffer
b. Pipeline
2. Replication
a. General Multiprocessor -- Distribute, Memory, ALU, CU
i. Recentralize Memory
ii. Recentralize ALU
iii. Recentralize CU -- basis for ILLIAC IV
3. Both
C. ILLIAC IV is a Vector Processor

II. Hardware Structure

A. Organization Chart
B. ILLIAC IV Array -- General Description
1. Control Unit (CU)
2. Processing Element (PE)
3. Data Paths
a. Control Unit Bus (CU Bus)
b. Common Data Bus (CDB)
c. Routing Network
d. Mode Bit Line
C. An Illustrative Problem
1. DO 10 I = 1, N
10 A(I) = B(I) + C(I)
a. $N = 64$
b. $N < 64$
c. $N > 64$
2. DO 10 I = 2, 64
10 A(I) = B(I) + C(I-1)
a. Skew at Compile Time
b. Skew at Execution Time
3. DO 10 I = 2, 64
A(I) = B(I) + A(I-1)
D. ILLIAC IV Array -- A More Refined Description
1. PE
a. RGD
2. PU
3. CU
a. ADVAST
b. FINST
c. MSU
d. TMU
e. ILA
E. Another Illustrative Problem
1. Laplace's Partial Differential Equation describing the Steady-State heat distribution on a slab
F. Data Allocation

- G. ILLIAC IV I/O System
  - 1. I/O Subsystem
    - a. CDC
    - b. BIOM
    - c. IOS
  - 2. Disk File System
  - 3. B6500
    - a. CPU
    - b. Memory
    - c. Multiplexor
    - d. Peripherals
      - i. Remote Terminals
    - e. Laser Memory
    - f. ARPA Network

III. Configurations at CAC and Paoli

IV. Diagnostics and Test/Repair Equipment

- A. IDIAP
- B. PEX
- C. PEMX
- D. CUCT
- E. Some Physical Characteristics
  - 1. Slides

V. Programming Languages

- A. ASK
  - 1. Background, Review, Notation, Conventions
  - 2. Sample Problems
    - a. Summing an Array of Numbers
    - b. Finding the Maximum Value in an Array of Numbers
    - c. Matrix Multiplication
      - i. Skewed Storage
    - d. Temperature Distribution on a Slab
      - i. Case 1: one temperature value per PEM
      - ii. Case 2: eight temperature values per PEM
- B. GLYPNIR
- C. FORTRAN

VI. Operating System

- A. ICL
- B. Utilities

VII. Some Applications

ADMINISTRATION

11.17 Administration and Services

11.17.1 Financial Report

Actual expenditures and obligations for January-March 1971:

	January	February	March
Burroughs Corporation	$312,000.00	($109,000.00)	N.A.[*]
University of Illinois	183,220.34	240,118.42	$168,093.51

Expenditures and obligations to date through March 1971:

Burroughs	$26,723,000.00
University	6,814,273.83

Budgeted expenditures - 3rd quarter, fiscal 1971.

	January	February	March
Burroughs	$567,900.00	$298,643.00	$252,758.00
University	215,302.00	215,302.00	215,302.00

[*] Monthly status report not received from Burroughs as of 4/30/71, therefore figures for March are not available.

REFERENCES

[1] Rajan, S. "Numerical Solution of the Partial Differential Equations of Gas Dynamics." Center for Advanced Computation, University of Illinois at Urbana-Champaign, (in preparation).

[2] Richtmyer, R. D., and Morton, K. W. "Difference Methods for Initial Value Problems." Interscience Publishers, 1967. Pp. 354-365.

[3] Rutishauser, H. "The Jacobi Method for Real Symmetric Matrices." Handbook Series Linear Algebra, Numerische Mathematik 9, 1-10, (1966).

[4] Eberlein, P. J., and Boothroyd, J. "Solution to the Eigenproblem by a Norm Reducing Jacobi Type Method." Handbook Series Linear Algebra, Numerische Mathematik 11, 1-12, (1968).

THESES

Moreno, V. "A Logic Test Generation System using a Parallel Simulator." Master's thesis. Urbana, Illinois: Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

Parker, J. L. "Logic per Track Information Retrieval System." Ph.D. thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

DOCUMENTS

Hashimoto, A., and Stevens, J. "Path Cover of Acyclic Graphs." ILLIAC IV Document No. 239. ILLIAC IV Project, University of Illinois at Urbana-Champaign, (December 24, 1970).

Schuster, Stewart. "An Information Management and Analysis System for ILLIAC IV." CAC Document No. 1. Center for Advanced Computation, University of Illinois at Urbana-Champaign, (December 11, 1970).

Schuster, Stewart. "A Statistical System for ILLIAC IV." CAC Document No. 2. Center for Advanced Computation, University of Illinois at Urbana-Champaign, (December 11, 1970).

Stevens, James, Jr. "Matrix Multiplication Algorithms for ILLIAC IV." ILLIAC IV Document No. 231, Department of Computer Science File No. 855. ILLIAC IV Project, University of Illinois at Urbana-Champaign, (August 26, 1970).

12. GENERAL DEPARTMENT INFORMATION

12.1 Personnel

The number of people associated with the Department in various capacities is given in the following table:

	Full-time	Part-time	Full-time Equivalent
Faculty	20	3	21.77
Visiting Faculty	4	0	5.00
Research Associates	0	0	---
Graduate Research Assistants	2	62	33.00
Graduate Teaching Assistants	0	28	13.11
Professional Personnel	21	0	21.00
Administrative and clerical	25	0	25.00
Nonacademic Personnel (Monthly)	54	0	54.00
Nonacademic Personnel (Weekly)	0	91	37.50
TOTAL	127	184	210.38

*This report does not include personnel employed on the ILLIAC IV Project.

The Department Advisory Committee consists of Professor J. N. Snyder, Head of the Department, Professors E. K. Bowdon, D. F. Cudia, K. W. Dickman, H. G. Friedman, C. W. Gear, D. B. Gillies, D. J. Kuck, B. H. McCormick, S. Muroga, T. A. Murrell, J. Nievergelt, J. R. Phillips, W. J. Poppelbaum, S. R. Ray, E. M. Reingold, J. E. Robertson, P. E. Saylor, D. L. Slotnick, and D. S. Watanabe.

12.2 Bibliography

During the first quarter, the following publications were issued by the Laboratory:

Report Numbers

(1) Ibaraki, T., T. K. Liu, D. Djachan, and S. Muroga, "Synthesis of Optimal Networks with Nor and Nand Gates by Integer Programming," Report No. 427, January 1971.

(2) "ILLIAC IV Quarterly Progress Report--October-December 1970," Report No. 431, January 15, 1971.

(3) Koo, Ping L. and Daniel E. Atkins, "Arithmetic Unit of ILLIAC III: Simulation and Logical Design-Part II," Report No. 418, October 28, 1968; Revised and Edited by Lakshmi N. Goyal, November 10, 1970.

(4) Maruyama, Kiyoshi, "Parallel Methods and Bounds of Evaluating Polynomials," Report No. 437, March 1971.

(5) McCormick, B. H. and B. J. Nordmann, Jr., D. E. Atkins, R. T. Borovec, L. N. Goyal, L. M. Katoh, R. M. Lansford, J. C. Schwebel, and V. G. Tareski. "ILLIAC III Reference Manual Volume I: The Computer System, Report No. 433, February 17, 1971.

(6) McCormick, B. H. and B. J. Nordmann, Jr., D. E. Atkins, R. T. Borovec, L. N. Goyal, L. M. Katoh, R. M. Lansford, J. C. Schwebel, and V. G. Tareski, "ILLIAC III Reference Manual Volume II: Instruction Repertoire," Report No. 434, February 26, 1971.

(7) Read, John, "Show-and-Tell, An Interactive Programming System for Image Processing System Specifications," Report No. 429, February 18, 1971.

Theses

(1) Adams, Harold Corwin II, "Dynamic Partitioning in the Array Language OL/2," (M.S.), Report No. 421, January 1971.

(2) Chan, Vivian Wan-Man, "Overlay Supervisory Programs," (M.S.), Report No. 419, February 1971.

(3) Latch, John Lockhart, "An Algorithm for Evaluating Array Expressions in OL/2," (M.S.), Report No. 422, January 1971.

(4) Mark, Barbara Drahos, "Machine Independent Compilation of PL/1: Pass I Symbol Manipulation," (M.S.), Report No. 439, June 1971.

12.2 Bibliography (cont.)

(5) Marshall, Larry Gene, "The Minicomputer: An Educational Tool," (M.S.), Report No. 432, February 1971.

(6) Muraoka, Yoichi, "Parallelism Exposure and Exploitation in Programs," (Ph.D.), Report No. 424, February 1971.

(7) Oberbeck, Peter Ernst Rudolf, "ORBIT--Online Reduced Bandwidth Infromation Transmission," (Ph.D.), Report No. 430, February 1971.

(8) Reingold, Edward Martin, "On Some Optimal Algorithms," (Ph.D.), Report No. 428, January 1971.

(9) Shiau, Lih-Er, "Design of Optimal One-Bit Adder Networks By Interger Linear Programming," (M.S.), Report No. 425, January 15, 1971.

(10) Taranto, Richard Joseph, "Numerical Studies of Stone's Symmetric Factorization and the Iteration Parameters, α and ," (M.S.), Report No. 423, January 1971.

(11) Winterbauer, Albert Lee, "PLONE: A Load-and-Go Compiler For A Basic Subset of PL1," (M.S.), Report No. 426, January 15, 1971.

(12) Yamada, Hirohide, "Emulation of Disc File Processor," (M.S.), Report No. 436, June 1971.

12.3 Colloquia

"On the Use of Functional Reasoning in Software Design," by Dr. Peter Freeman, Carnegie-Mellon University, Pittsburgh, Pennsylvania, February 8, 1971.

"The Anatomy of a Microprogrammed Communication Processor," by Professor James E. Vander Mey, Computer Science Department, The Pennsylvania State University, University Park, Pennsylvania, February 11, 1971.

"An Application of Automata Theory to the Isomorphism Problem for Planar Graphs," by Professor John Hopcroft, Computer Science Department, Stanford University, Stanford, California, February 15, 1971.

"Transducing Automata and Formal Languages," by Dr. Johannes J. Martin, 18835 Apache Drive, South Bend, Indiana, February 22, 1971.

"A Practical Model for Generating Code," by Mr. Thomas Wilcox, Department of Computer Science, Cornell University, Ithaca, New York, February 25, 1971.

"The Future of Instructional Computer Use," by Dr. Roger E. Levien, The Rand Corporation, 2100 M Street, N. W., Washington, D.C., March 8, 1971.

"Data Compression - Tradeoffs and Implications," by Mr. Ross Overbeek, Department of Computer Science, Pennsylvania State University, State College, Pennsylvania, March 11, 1971.

"Conversational Systems Programming," by Mr. Rudolph Krutar, Carnegie-Mellon University, Department of Computer Science, Schenley Park, Pittsburgh, Pennsylvania, March 15, 1971.

"Computing Polynomials with a Minimum Number of Operations," by Professor Albert R. Meyer, Project MAC, 545 Technology Square, Cambridge, Massachusetts, March 22, 1971.

"Learning to Use Contextual Patterns in Language Processing," by Mrs. Sara Jordan, Computer Science Department, University of Wisconsin, Madison, Wisconsin, March 25, 1971.

"Laser Displays and Related Research at GTE Laboratories," by Mr. Vernon J. Fowler, Bayside Research Center of GTE Laboratories Incorporated, Bayside, New York, March 29, 1971.

12.4 Drafting

During the first quarter, a total of 232 drawings were processed by the general departmental drafting section:

Large Drawings	67
Medium Drawings	16
Small Drawings	86
Layouts	9
Report Drawings	20
Changes	22
Miscellaneous	12
Total	232

(M. Goebel)

12.5 Shops' Production

Job orders processed and completed during the first quarter of 1971 are as follows:

	AEC 2118	AEC 1469	Other
Machine Shop	2	7	3
Electronics Shop	2	74	31
Chemical Shop	2	63	6
Layout Shop	1	56	9

(F. P. Serio)

COO-1469-0193
COO-2118-0015

QUARTERLY TECHNICAL PROGRESS REPORT

April, May, June 1971

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

QUARTERLY TECHNICAL PROGRESS REPORT

April, May, June 1971

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois

TABLE OF CONTENTS

Page

Page

1. CIRCUIT RESEARCH

(Supported in part by the Office of Naval Research under Contract N000 14-67-A-0305-0007, W. J. Poppelbaum, Principal Investigator.)

1.1 Bundle Processing (Project No. 21)

1.1.1 Bundle Signal Repeater and Restorer

During this quarter the design of the signal restoring section of the system was completed. Currently work is being done on the input portions of the bundle repeater and that of the bundler restorer. The signal readout section for both the repeater and restorer is also being designed.

Construction of the system will begin during the next quarter.

Bernard Tse

1.1.2 SABUMA

The final additions to the system have been designed and are being built. These are the error detecting circuits and their sequencing circuitry.

The bundles have been built and are being installed. The only job remaining, prior to the system being complete, is to install the display panel.

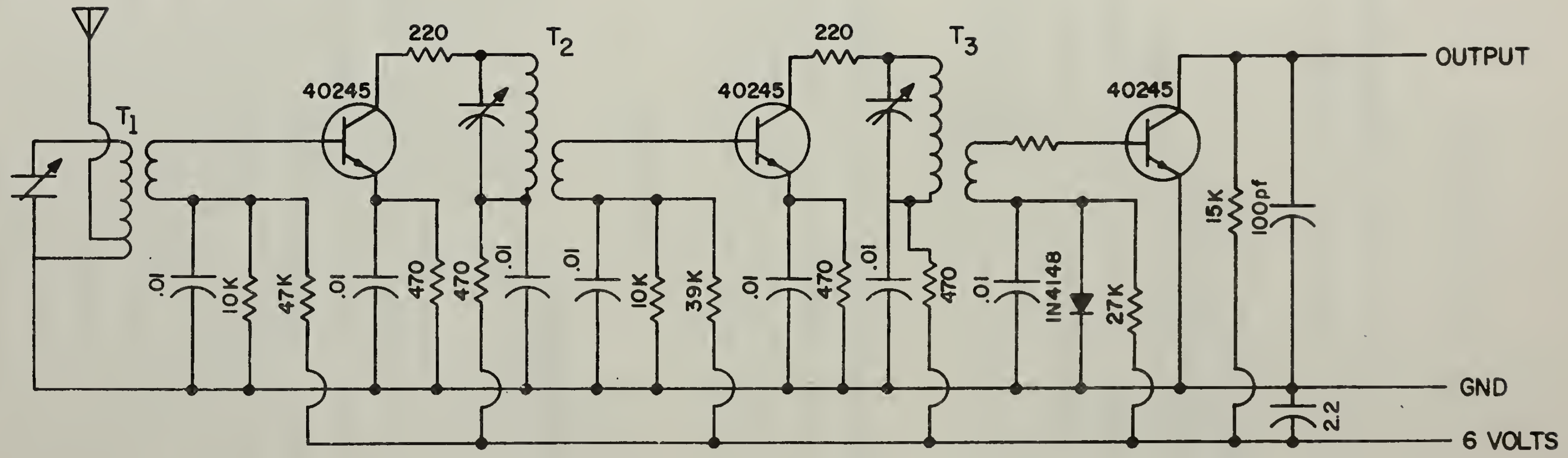
Trevor Mudge

1.2 APE (Project No. 25)

1.2.1 The 42.5 MHz Clock Receiver

Over the past quarter, much effort was devoted to the development of AM receivers for the APEs. In the design of these receivers, the overriding factor is low power consumption. This precludes the use of some newer devices, such as dual gate MOSFETs, as the active elements. Bipolar junction transistors are used in both the clock receiver and the data receiver.

Figure 1 shows the schematic diagram of the 42.5 MHz receiver. It is a two-stage tuned RF amplifier followed by an AM detector. The carrier frequency of 42.5 MHz is chosen to be much higher than the band of frequencies for the data channels. Furthermore, the subharmonics of the clock carrier frequency also lie outside this band. The sensitivity of the clock receiver is about 500 μV. It draws 2.5 mA from a 6 volt power supply, and has a 1 MHz bandwidth to receive pulses as little as 2 μs wide. Due to the binary nature of the clock signal and the careful frequency assignment to avoid



T_1 : HEATH GD-19-40-916 TRANSFORMER

T_2 and T_3 : DALE IPB-2042-31 RF TRANSFORMERS

Figure 1. The 42.5MHz Clock Receiver

interference, automatic gain control is not necessary for the clock receiver. RCA transistor 40245 was chosen for the clock receiver because of its exceptionally high gain at a low bias level. Although it is not shown in the schematic, provision is made for the insertion of a neutralization capacitor between the collecter tank circuit and the base of the transistor in the tuned RF stages. The divider in the base circuit of the detecting stage is to bias the transistor just below the threshold of conduction. When the 42.5 MHz signal appears at the input to the clock receiver, the detecting transistor will be driven into saturation and will remain there as long as the 42.5 MHz carrier frequency appears at the input to the clock receiver. Since the 42.5 MHz carrier is switched on and off according to the clock signal, the output from the clock receiver produces an inverted clock signal.

Yiu Kwan Wo

1.3 PENTECOST (Project No. 31)

1.3.1 Testing the Land System for Sequential Presentation

Two slide projectors were used to project the red and green fields of the Land Color system. A plastic wheel, half of it painted black, was rotated by a synchronous motor at 1800 rpm. The wheel was placed betwen the two projectors. By rotating the wheel the fields were successively projected on the screen at a field frequency of 30 cps. The Land effect was still observed with this sequential presentation. The proper combination of filters to give the best color effect is now being investigated.

The camera for the Pentecost system has been ordered from General Electric. An Amperex red extended Plumbicon was ordered for this camera.

G. Panigrahi

2. HARDWARE SYSTEMS RESEARCH

(Supported in part by the Atomic Energy Commission under Contract US AEC AT(11-1) 1469, W. J. Poppelbaum, Principal Investigator.)

2.1 LASCOT (Project No. 09)

2.1.1 Laser Specifications

The specifications of the laser needed for LASCOT have been derived. A mixed gas (argon-krypton) is needed, with an output of 200-250 mW in both blue (4880A°) and red (6471A°) and 75 mW in the green (5145A°). A survey of the available lasers has been made.

M. N. Cooper

2.2 OLFT

2.2.1 Intermediate System Tests

An intermediate version of the cooled-crystal light-valve has been constructed and consists of the entire cooling assembly of the new system mounted in the vacuum chamber of the old system, using the old electronics and optics. This facilitates testing of the cooling components - sample crystal and substrate, thermoelectric elements, heat sink, temperature sensors and control circuits - under conditions independent of the new electron beam system. This intermediate system can be used only in a single-frame mode, requiring manual frame gating and erasure.

Initial testing has begun on this system, and the results have led to several conclusions. The first cooling cycle was manually controlled, at a rate of about 20°C per hour. Unfortunately, the substrate cracked at -6°C. Although this test substrate is not mounted exactly as planned for the new system, it is obvious that this is a critical problem. Also, to minimize stress, the temperature cycle should require on the order of several hours for complete cooling or warming. The temperature control circuits are now designed for steady-state operation, and should be redesigned for automatic control of the temperature cycle.

Fortunately, the substrate damage affected only a corner of the crystal, so that tests will continue, specifically covering crystal time constant, uniformity of crystal characteristics, assembly aging and surface damage, secondary emission, etc. This should result in predictable performance when the new crystal assembly is constructed.

Douglas Sand

2.3 ORBIT (Project No. 15)

2.3.1 Receiver Synchronization

The low bandwidth synchronization scheme, designed by Ed Carr, which has been discussed in its various phases in these more recent reports, has been built and is now working. A full description of the design and its implementation will shortly appear as an M.S. thesis. This brings the ORBIT project, therefore, to its conclusion.

M. Faiman (ed.)

2.4 Tricolor Cartograph (Project No. 16)

2.4.1 Physical Wiring

The switches and potentiometers have been wired into the control panel and connections made in the display junction box to allow for them. Work was done on the lightpen to make it pin for pin compatible with the Artrix pen. A new pen enable circuit was installed. Wiring was also done in the main junction box. A platform was built for the Corning delay line, which is now mounted in its place.

The first phase of the store outline circuitry described in the last report has been wired and works as expected.

2.4.2 Power Supply

Finally, a power supply was designed and tested for the Corning delay line. The schematic, shown in Figure 1, also shows the overvoltage protectors.

Don Hanson

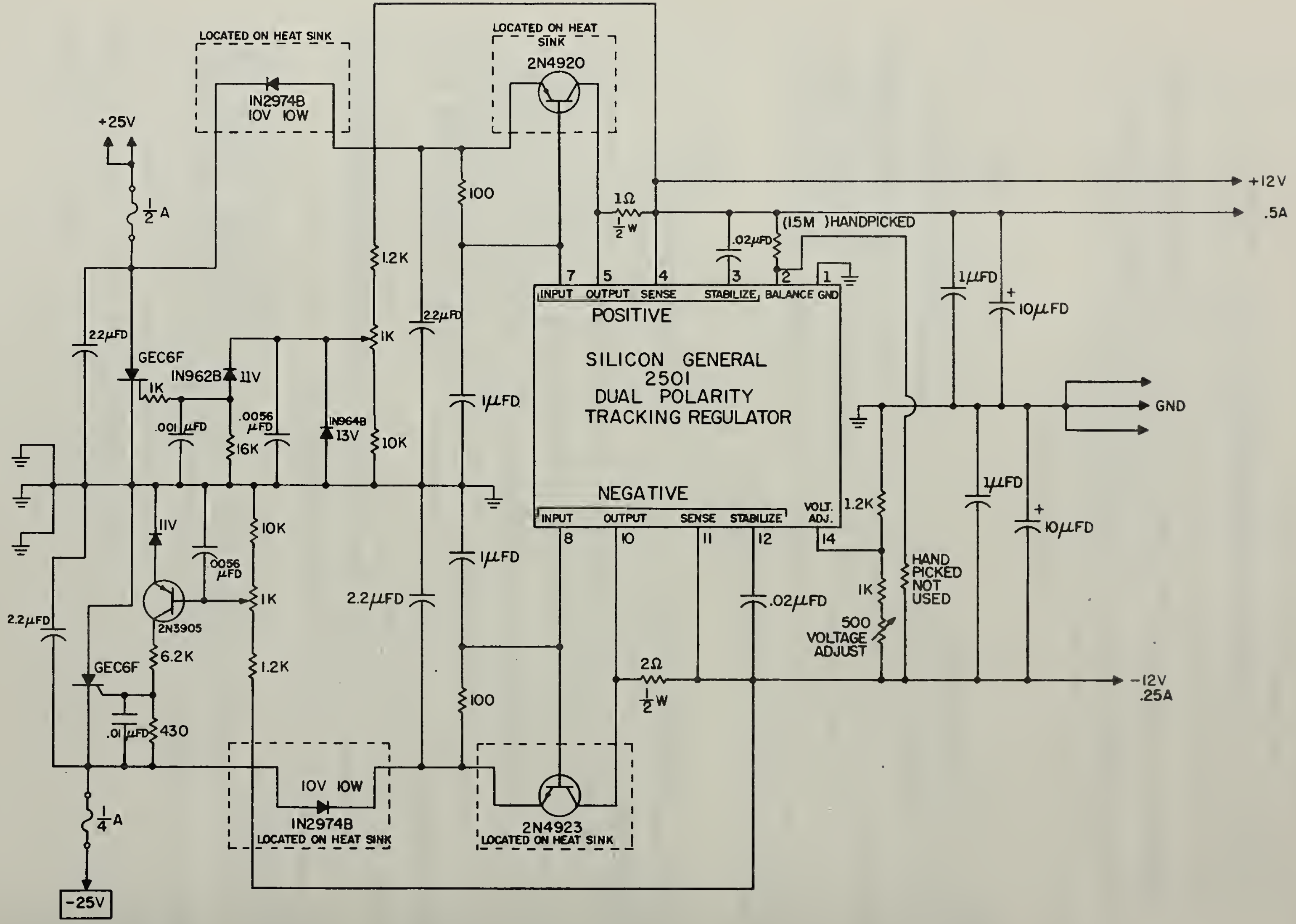


Figure 1. Delay Line Power Supply

2.5 BLAST (Project No. 19)

2.5.1 Screen Signal

As will be recalled from a previous report, the screen signal was to be picked up by two interwoven combs and amplified by a differential amplifier which was floating at the second anode potential (approximately 18 kilovolts). A reasonable amount of success was achieved using this method, as reported last time.

Unexpectedly, during the last quarter, the internal connections to one of the conducting combs open circuited. This precludes the use of the interwoven combs to obtain a synchronizing signal; although one comb is still connected it may break down at any time. The reason for the breakdown is not known, but it is conjectured that the "sandwich" type of connection used to contact the nickel conducting pad on the glass screen was constructed of incompatible metals causing a nonconducting layer to be formed between layers of metal. Therefore, a second approach to the problem of aligning and synchronizing the combined video information behind the vertically oriented lenses is being tried.

2.5.2 Scan Correction

It is certain that the monitor scanning pattern can be controlled so that the video information is aligned with the lenses. An initial attempt at this proved quite successful. Using seven surplus relay coils placed ahead of the yoke in close relation to the tube envelope, a static correction was applied experimentally to the raster. The correction was sufficient to provide alignment for perhaps, one-third to one-half of the raster. In this area the 3-D effect was quite striking to many observers. It is apparent, however, that satisfactory alignment over the entire raster can only be accomplished using a dynamic correction. To this end a diode shaping waveform generator has been constructed and coils are being driven dynamically to determine the actual correction needed. Moderate success has been achieved in this area. It is felt that this approach is almost certain to succeed and will by-pass the troubles that have plagued BLAST due to the manufacturing of the special tube with its conducting stripes.

Larry Wallman

2.6 Eidolyzer (Project No. 23)

Eidolyzer is finished; Art Simons has his Ph.D. The interested reader may peruse his thesis: "Eidolyzer: A Hardware Realization of Context-Guided Picture Interpretation," D.C.S. Report No. 448, June 1971.

Demonstrations of the machine may be had in Room 231, Digital Computer Lab (by appointment, please).

M. Faiman (ed.)

2.7 Semantrix (Project No. 24)

The circuit boards for the sense amplifiers have been constructed and are under test. The control logic has been designed and fabrication is under way. This will be tested out upon the receipt of a teletype now on order.

Trevor Mudge

2.8 LINDA (Project No. 28)

The goal of this project is to build a LINe Drawing Analyzer capable of recognizing a number of simple line drawings. A flying spot scanner is used to project a given drawing on an oscilloscope. Photocells are used to extract edge information from the drawing. The information from the photocells will be decoded to produce an identifying output.

2.8.1 Project Status

Past work has been focused on building a circular array of photocells, expanding the pattern to the circle's edge, and trying to find a simple decoding scheme. A flying spot scanner was obtained in January and deflection circuits and a video amplifier have been built. Expansion of the figure was originally planned to result from the lowering of the accelerating voltage across the CRT. Work this past quarter indicates this method is not the best since focusing and intensity problems cannot be conveniently overcome. Expansion of the figure will be accomplished by raising the deflection voltage of the flying spot scanner - work on this has not yet begun.

Due to lack of a simple decoding scheme for the circular array, work was temporarily stopped and new work (particularly during the last quarter) was begun on a linear array.

The principal of the linear array is illustrated in Figure 1, using a simple triangular figure. The triangle is swept across the photocell line, turning on every photocell which comes into its boundary. The sum of the photocell signals is taken and is termed the ordinate sum. The ordinate sum signal is differentiated twice and, ideally, the second differentiation produces one pulse for each discontinuity in the figure. This analog decoding scheme provides a simple method of recognition for polygons by counting pulses.

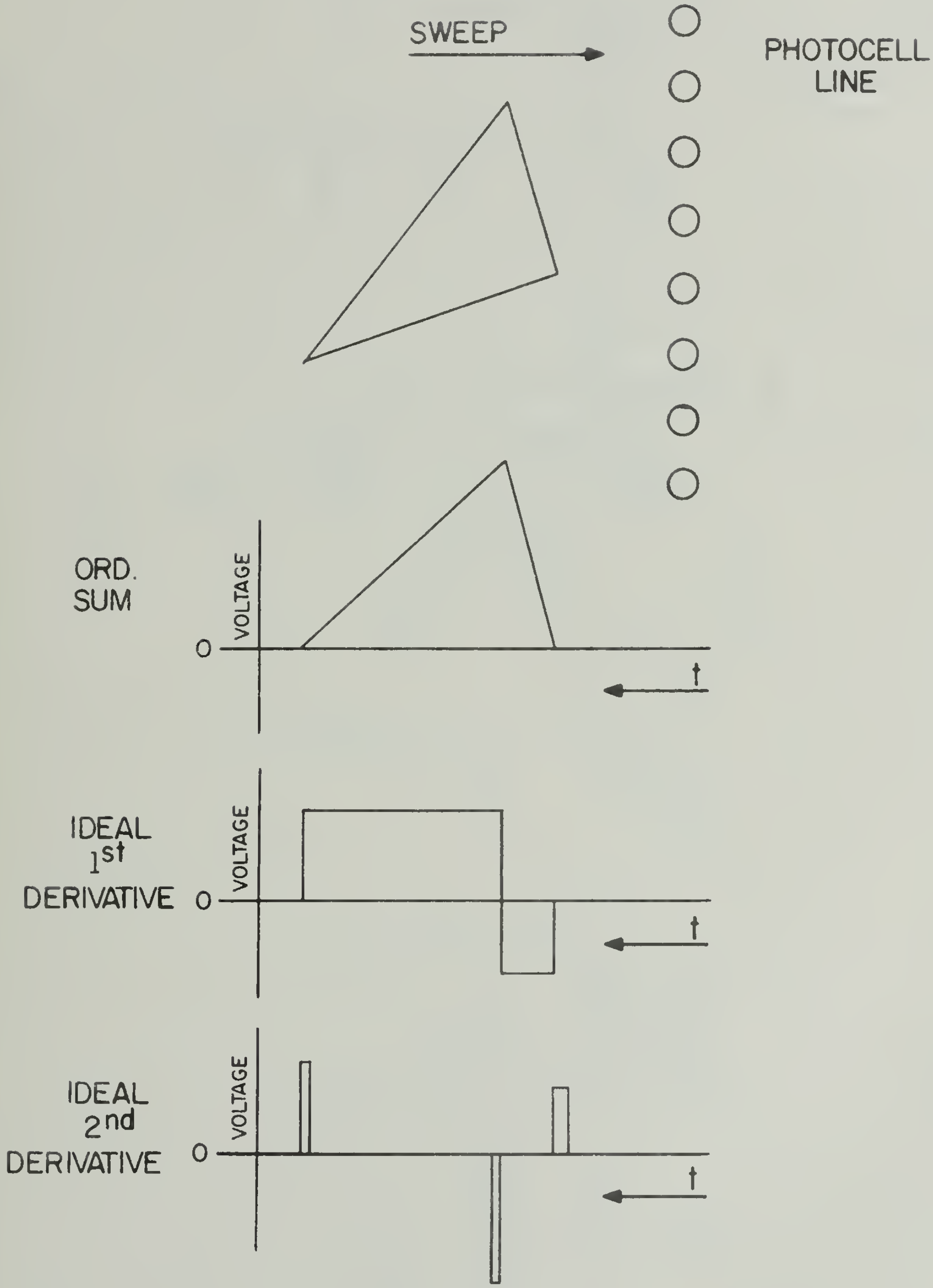


Figure 1. Illustrating Principle of Linear Array

To date a linear array consisting of 10 photocells has been built and tested. Figure 2 shows the circuit used to sweep the pattern across the array. Results have been promising but not entirely satisfactory. The photocells used have a 1/5" diameter and do not provide sufficient resolution. The ordinate sum signal is not smooth, and it is difficult to distinguish a 5 from a 6 sided figure.

2.8.2 Future Work

Two techniques will be tested to improve resolution. First, fiber optics will be employed. Small fiber bundles can be grouped closely in a linear array to carry light to photocells which are less closely spaced. Second, two or more photocell lines can be staggered, as shown in Figure 3. The signal from the first line is delayed, so that all photocells appear to be in the same line. The fiber optics technique will be tried first. The delayed array will be used if needed. It is hoped that patterns having up to 15 sides may be distinguished.

Dick Blandford

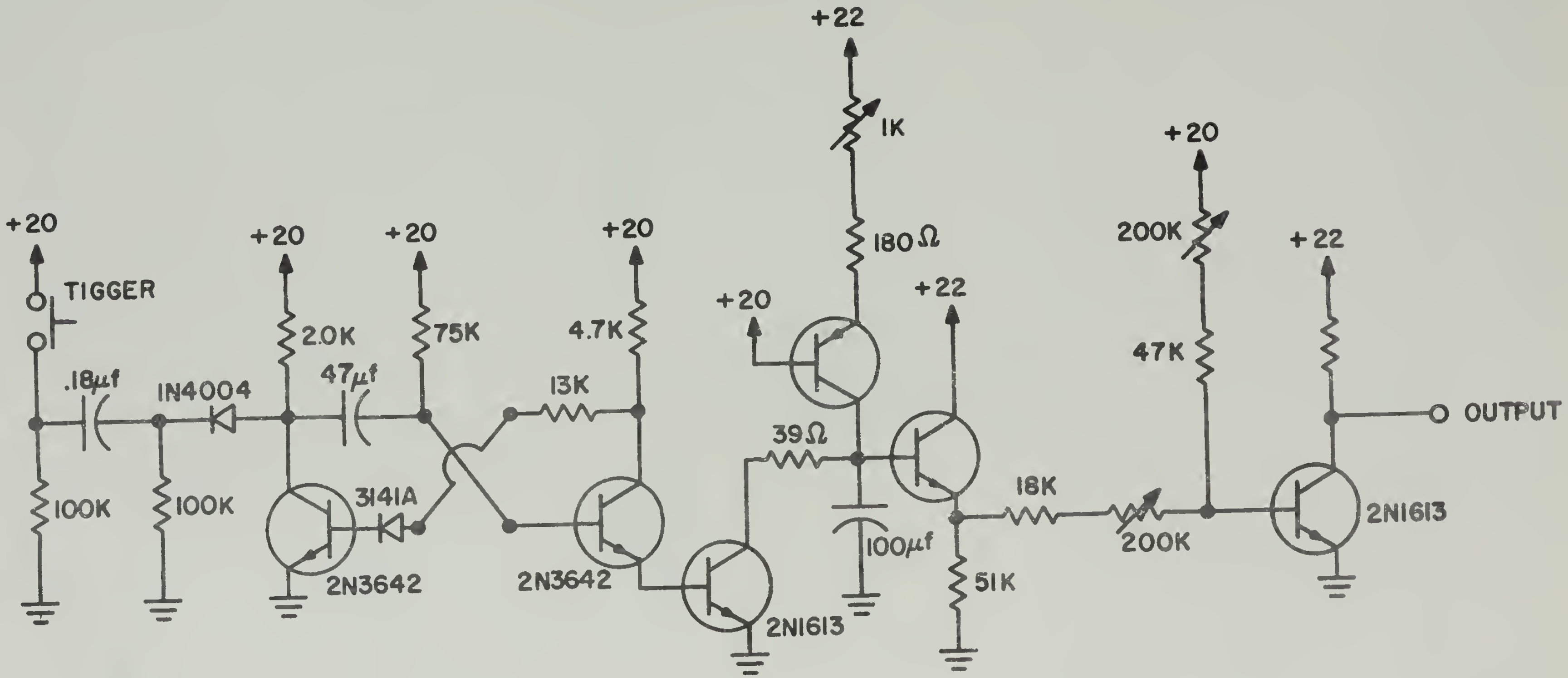


Figure 2. Circuit to Sweep Pattern Across Linear Photocell Array

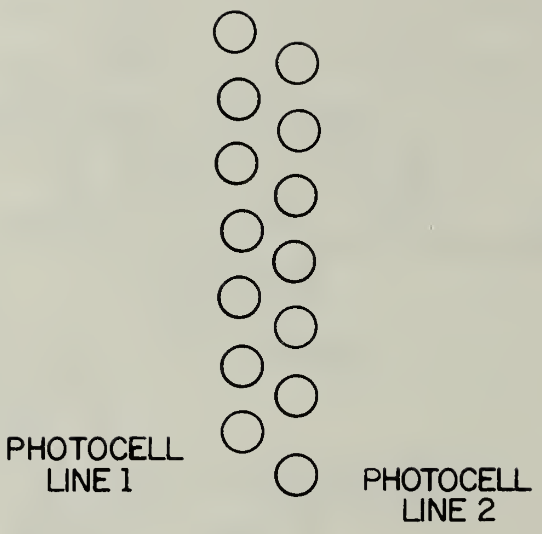


Figure 3. Staggered Photocell Lines

2.9 Stereomatrix (Project No. 30)

2.9.1 Transformer

All the cards for the transformer has been tested. The rack wiring is done and debugged. The transformer has been tested and run for several hours.

The outputs of PAGAN, the three-dimensional pattern generator, have been used as input to the transformer to see two different pictures (one for each eye). The transformer output was connected to a CRT and a change in perspective was observed by a simulated moving of the observer's position.

Some changes in the cards of the perspective generator have to be made so that output of transformer is compatible with the deflectors. It is hoped to accomplish this in the next quarter.

Shiv Verma

2.9.2 Display

Some further refinements on the laser power supply were made. The low-voltage control board was removed from the supply and driven with a pulse generator to determine its transient response. As a result of this study a capacitive divider was removed from the control board. Also, a heat sink and diode isolator were added to the 22-volt zener on the control board. No transistor or zener failures have yet occurred with this modified circuit. Remaining to be changed is the start relay section of the control board. This section was previously changed from a differential amplifier to a monostable. However, large line transients caused by nearby motor start-up and some control settings still incorrectly trigger the start transformer on.

Two ultrasonic deflectors have been received and assembled into a two-dimensional display. Geometric line patterns with no blanking have been displayed using signals from PAGAN. The patterns are reasonably bright in a darkened room and dimly visible with normal room light. In this test set-up the laser light encounters eight uncoated lenses and two coated deflectors. The extinction is excellent for a crossed polaroid. This is using a temporary aluminized lenticular front projection screen.

Simple emitter follower circuits were designed to convert the input ±10 volt signals to the -0.5 to -7.0 and -1.5 to -14.2 voltage levels required by the ultrasonic cell drivers. As the cells are rather non-linear, more sophisticated conversion amplifiers are being designed.

A third deflector has subsequently been received and the first unit is being returned for repair. The first unit developed a frequency jump problem. It is not as efficient as the second and third units either. As soon as it is returned the complete three-dimensional system will be assembled.

Steve Whiteside

2.9.3 Observer Position Detector

All digital circuit boards have been designed, built and tested.

The encoder assembly in its entirety has been built and some success has been achieved in reading the disc optically. Variations in the manufacture of phototransistors have altered their alighment with the optic fibers in the reader mount to the extent that some tracks cannot be detected. Also several sources are being considered which should provide uniform illumination to the source optic fibers.

During testing, some crosstalk betwen optic channels in the reader has been experienced. Previously .040" source light fibers were replaced with .020" source fibers to reduce crosstalk. It may be necessary to use .020" fibers also on the pickup side of the disc to insure proper channel separation. In the case where .020" fibers are installed, spacers of the same size are provided to separate and align them.

During the next quarter, the encoder readout will be improved and tested with the detector circuitry. The second reader assembly will be built and the analog circuitry constructed.

Chuck Pirnat

2.10 PAGAN (Project No. 32)

2.10.1 Completion of the Project

The PAGAN project (PAttern Generator ANalog) reached completion during this quarter. The project has been documented in the form of a thesis. Further documentation is now being assembled into an operating manual.

At the beginning of the second quarter, three of PAGAN's circuits had been laid out on printed circuit cards. As much of the remaining circuitry was not highly repetitive, in the interests of economies of time and resources, most of those circuits were handwired on universal cards.

One circuit that is used a number of times was laid out on a printed circuit card. This card is the 8 x 10 programming matrix for storage of preset patterns. PAGAN's patterns may be specified by entering the appropriate digital increments and limits on front panel switches. As a convenience to users who do not wish to derive the proper control signals, sixteen preset patterns are stored internally. The control signals for the preset patterns are stored on the 1469 - 491 card shown in Figure 1. The inputs of the TTL gates are used in place of the diodes in a diode matrix. The vertical and horizontal lines of the figure are on opposite sides of the printed circuit card. This wired - in storage is programmed by inserting a pin at a location indicated by an open circle and soldering it to the printed circuit on each side.

2.10.2 Suggestions for Improvements

The PAGAN project has met its desired goals as a pattern generator for refreshed display of three-dimensional geometric figures. The present circuitry is operating satisfactorily and is not in need of modification. However, the project can be extended, for example to include spherical and toroidal surfaces. If additions are considered, some circuit improvements might be contemplated.

If some of the hand wired digital circuits are to be converted to printed circuit layouts, use should be made of some of the newer MSI TTL devices. When changing from hand wired univeral cards, a different partitioning of the circuitry could clear some of the congestion of the card cage back panel wiring.

It would be easier to adjust the analog circuitry for optimum performance if some modifications were made. The Sinusoidal Transconductors require ±15V accurate to .01%. Currently, the sinusoidal waveforms are tuned up by adjusting the system's ±15V supplies, which also affects the other analog modules. Any redesign of the 1469 - 485 card probably should include the addition of Integrated Circuit voltage regulators.

The card containing the analog switches has undergone some slight modifications this quarter. These switches are made up of discrete components and Junction FETs. If the circuit should require future adjustments, consideration should be given to incorporating Integrated Circuit MOSFETs.

Richard Partridge

1469-491

8 X 10 Programming Matrix

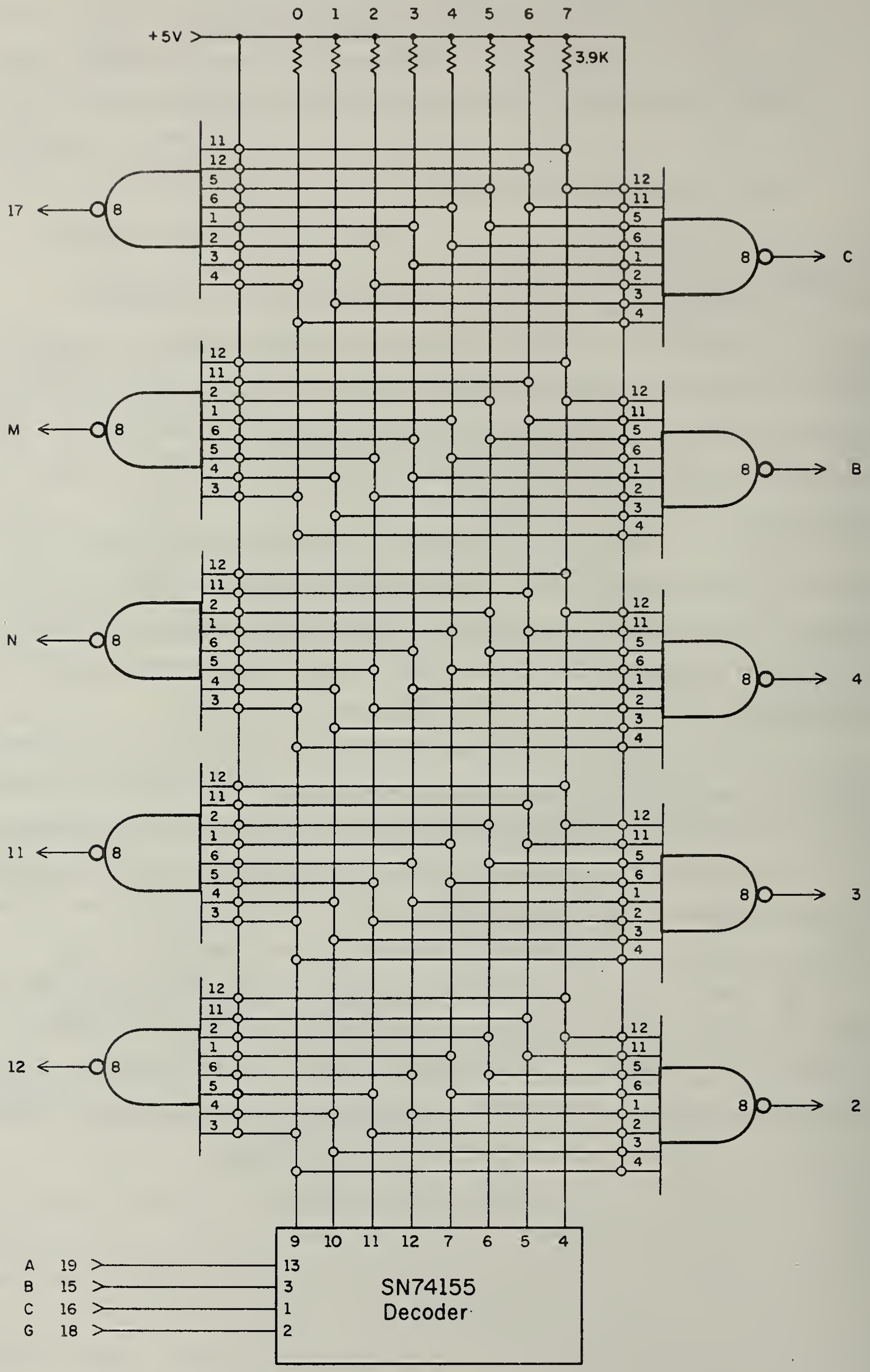


Figure 1. Programming Matrix Card

2.11 Scantrix (Project No. 35)

2.11.1 Possible Implementations

Four possible schemes for Scantrix were considered. The first one utilizes a rotating mirror to reflect light from a line of 128 light emitting diodes (LEDs), thus scanning the picture across a screen. The second approach(Figures 1 and 2) has four lines of LEDs instead, mounted on the surface of a cylinder, and with a 90-degree angle between each line. As the cylinder rotates, signals are fed into the lines of LEDs sequentially, to scan out the picture, as in the first method. However, the second approach, with four lines of LEDs, eliminates the jumping of the scanning device (the mirror in the first case) since the diodes are now oriented so that when one line reaches the bottom of the screen the next line will just come in at the top. But, both of these two approaches have to make use of fast rotating motors to rotate the scanning structures which are reasonably big and probably heavy. Besides, high intensity is required for the LEDs in order to give reasonably bright project.

The other two approaches require 128 x 128 LEDs to form the whole frame, the advantage of these electronic scanning approaches over the mechanical scanning approaches aforementioned should be obvious.

Refer to Figure 3 for the analog switching approach. In this scheme, the video signal from the TV is sampled at appropriate intervals and directly fed into the frame of LEDs through control gates. The requirement of gray levels and switching over the 128 x 128 (16,384) LED elements presents a financial problem, since a switch has to be incorporated with each of the LEDs in order to light each up in sequence in accordance with the TV scanning structure.

The last scheme uses digital switching, which can be economically achieved, made possible by the low price of digital ICs. Refer to Figure 4 for a simplified block diagram of the scheme. The video signal from the TV is sampled at appropriate intervals (128/line), as before. The sampled signals, instead of being fed directly into the LEDs, are changed to digital signals via an A/D converter. Every 128 sampled, digital signals, corresponding to one single line on the screen, are stored in a buffer. There are two buffers alternating in storing the signals. One is being fed with the signals while the other is being gated to the next stage, namely, 128 D/A converters. Thus the original signals are recovered and are used to drive the LEDs, one line at a time.

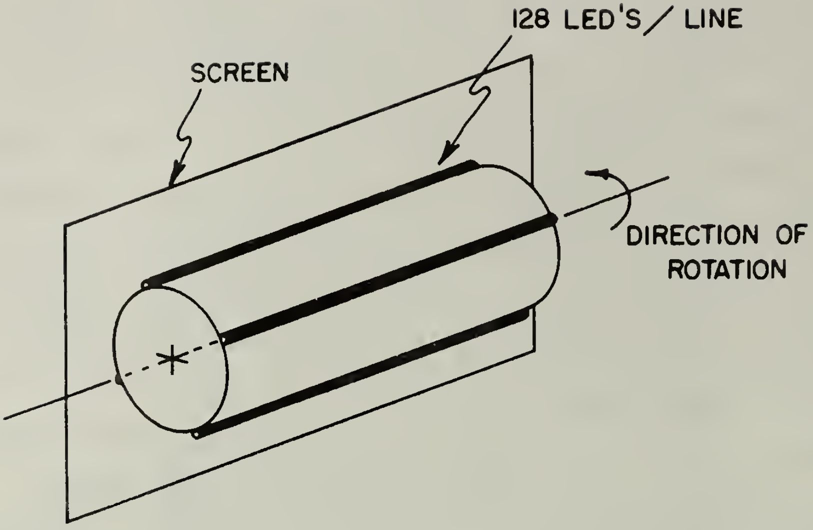


Figure 1. Rotating Drum with Four Lines of LEDs

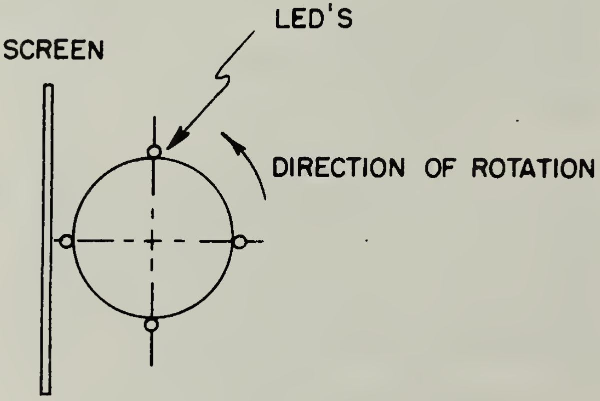


Figure 2. Side View of Rotating Drum

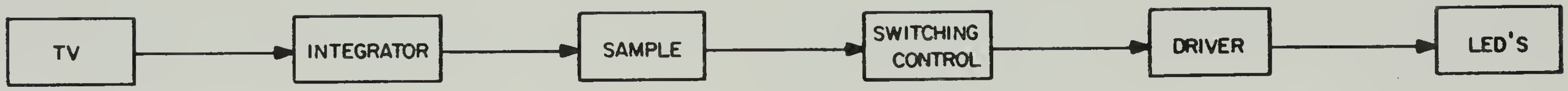


Figure 3. Block Diagram of Analog Switching Approach

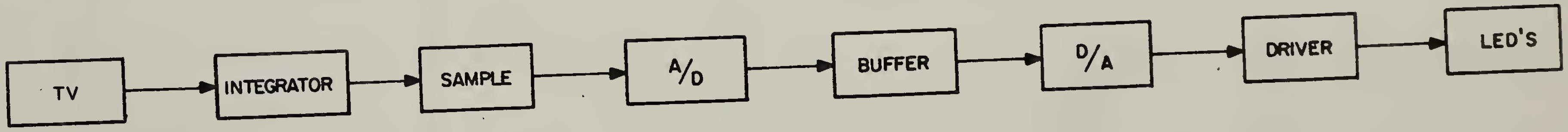


Figure 4. Block Diagram for Digital Switching

A closer look at the system reveals that an integrator will be needed to average out the video signals along each of the 128 intervals of the 62.5μs in between every horizontal blanking, in order to produce a reasonable value for the sampling circuit. Direct sampling is avoided because in each of the 128 intervals (very roughly 0.4μs/interval) drastic differential change in the video signal can occur for a very short period of time. If integration is not provided, sampling at that particular instant when the spike occurs obviously gives a different value from that sampled at another instant. Since there are only 128 rows of LEDs on the screen, integration has to be made vertically also. However, this can be easily overcome by allowing our humans eyes to do the integration instead. In other words, appropriate numbers of lines are superimposed onto each row of LEDs (4:1, including interlacing).

Note that two buffers have to be used, each consisting of 128 x 4 bits (128 bits for 128 LEDs, 4 bits of gray levels for each point). One buffer is gated to drive the LEDs while the other is loaded at the same time.

Sik Yuen

Form AEC–427
(6/68)
AECM 3201

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY–TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO. COO 1469-0193	2. TITLE QUARTERLY REPORT - April, May, June 1971

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report

□ b. Conference paper not to be published in a journal:

Title of conference ________

Date of conference ________

Exact location of conference ________

Sponsoring organization ________

□ c. Other (Specify) ________

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.

□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.

□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

W. J. Poppelbaum, Professor of Computer Science and Electrical Engineering
Editor: M. Faiman

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature W. J. Poppelbaum	Date July, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.

□ b. Report has been sent to responsible AEC patent group for clearance.

□ c. Patent clearance not required.

3. SOFTWARE SYSTEMS RESEARCH

Publications this quarter:

Guimaraes, C. J. F. WEED: A Wonderful Equation Elimination Device, Department of Computer Science Report No. 444, University of Illinois, Urbana-Champaign, Illinois 61801; also, submitted in partial fulfillment for the degree of Master of Science in Computer Science, June 1971.

Ratliff, K. SPARSE MATRIX INVERSION, Department of Computer Science Report No. 443, University of Illinois, Urbana-Champaign, Illinois 61801, June 1971.

C. W. Gear, Professor
and Principal Investigator

3.1 Numerical Processes

3.1.1 Sparse Matrix Inversion (J. Deogun)

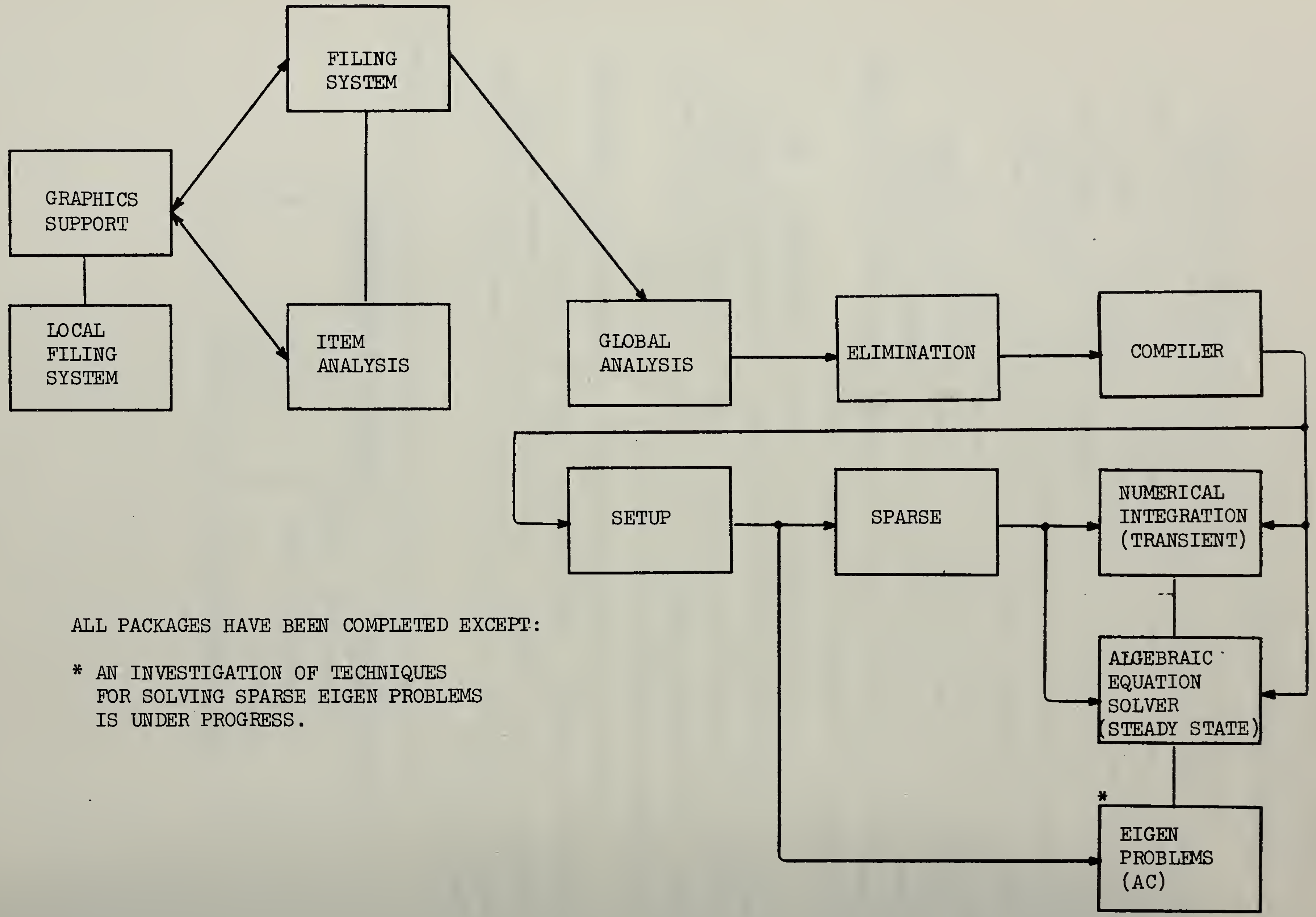
Sparse matrices occur frequently in engineering and industrial problems. Expert handling is needed to develop routines for (SPARSE) matrix operations. Currently we are concerned with the group of routines which generates matrix inversion code for the Jacobian of a system of equations. These routines have been combined with the main program and the subroutine DIFSUB via the subroutines P1 and P2. The function of DIFSUB is to integrate ordinary differential equations. These results are used in the corrector step of the integration process. The combination of all these subroutines is called the numerical package. Figure 1 gives the block flow diagram of the interaction of various processes which constitute the set of packages which perform the numerical and associated processing for transient analysis.

The subroutine SPARSE has been improved considerably. The section of SPARSE which selects pivot elements at each step has been successfully modified, and its present version is considered to have the "best" pivot selection algorithm. The meaning of the word "best" used in this context is explained in the latter part of this report.

Previously, various pivot selection strategies were tested only on small test examples. Recently, they have been tested with a practical (53 x 53) example. Various algorithms were compared and the basis for comparison is the number of instructions generated by subroutines MATMUL and MATINV.

Following are the results of this comparison on the 53 x 53 example:

No.	STRATEGY	NUMBER OF INSTRUCTIONS
1	Min Row--First Column	1547
2	Min Row--Min Column	1128
3	Min PRDT	1132
4	Min Column--Min Row	1076



ALL PACKAGES HAVE BEEN COMPLETED EXCEPT:

* AN INVESTIGATION OF TECHNIQUES FOR SOLVING SPARSE EIGEN PROBLEMS IS UNDER PROGRESS.

Figure 1

Pivot selection method 4 (i.e. Min Column--Min Row) has produced the best results with almost all examples. In this method a column with a minimum number of elements is taken as the pivot column. If there is more than one such columns, a chain of minimum columns is formed. Then elements in the minimum column(s) is scanned to choose a pivot. Each "eligible" element in the pivot column is compared by the number of elements in its row. An element is eligible for pivoting if it does not lie in either the row or column which has already been pivoted. It may be recalled here that all pivot selection strategies are dynamic; at each step the choice depends only on that part of the updated matrix which consists of unpivoted rows and columns. Choice of a pivot is also restricted by its size. An element may not be used as a pivot if it is less in absolute value than α times the largest element in its column. α is a constant. Values of 0.1 or 0.01 have been used in recent examples. Better results have been obtained for $\alpha = 0.1$.

The SPARSE program has been improved very much, particularly to minimize round-off error. The present version of pivot selection strategy, as described above, helped much in improving upon the round-off error. Subroutine check, a listing of which is given below, was added to the group of SPARSE routines. This subroutine calculates a relative error vector at various steps. It provided considerable use for minimization of round-off error.

```
          SUBROUTINE CHECK(DIN,DOUT,MX1,PW1,IRC,N,S,K1,L)
          IMPLICIT REAL*8(A-H,Q-Z)
          DIMENSION DIN(1),DOUT(1),MX1(6,1),PW(1),IRC(1)
          DIMENSION PW1(1)
          DIMENSION T(60),TMAX(60)
          K=K1
          S=0.
          DO 3 I=1,N
          T(I)=0.0
          TMAX(I)=0.D0
          J=IRC(I)
        2 IF(J.EQ.0)GO TO 3
          Q=MX1(6,J)
          IF(MX1(1,J).GT.1)Q=PW1(MX1(6,J))
          IF(MX1(1,J).EQ.0)Q=0.
          T(I)=T(I)+Q*DOUT(MX1(3,J))
          TMAX(I)=DMAX1(TMAX(I),DABS(T(I)))
          J=MX1(4,J)
          GO TO 2
        3 CONTINUE
        6 IF(K.EQ.0)GO TO 5
          I=MX1(2,K)
          J=MX1(3,K)
          T(I)=T(I)+DOUT(J)
          TMAX(I)=DMAX1(TMAX(I),DABS(T(I)))
          K=MX1(5,K)
          GO TO 6
        5 IF(L.EQ.0)GO TO 8
          DO 9 I=1,N
        9 DIN(I)=T(I)
          RETURN
        8 DO 7 I=1,N
          TMAX(I)=DMAX1(TMAX(I),DABS(T(I)-DIN(I)))
          IF(TMAX(I).EQ.0) GO TO 7
          S=S+DABS((T(I)-DIN(I))/TMAX(I))
        7 CONTINUE
          WRITE (6,4) S
    4     FORMAT(1H ,7HCHECKS=,D15.5)
        1 CONTINUE
          RETURN
          END
```

SPARSE in its present form is considered quite efficient. The combination of all subroutines in the numerical package has been run successfully with small test examples. Currently, the package is being tested with the 53 x 53 example. Except for minor changes in P1 and P2, the package runs successfully. Relation between various routines or sections of the numberical package are shown diagrammatically in Figure 2.

THE EIGENVALUE PROBLEM

Reasonably efficient routines are available for the solution of the standard eigenvalue problem $Ax = \lambda x$. Our present concern is the more general problem

$$Ax = \lambda Bx$$

where A and B are of sparse nature.

If A and B are both singular and of rank appreciably lower than their order, the problem can be handled by Wilkinson's method.* The SPARSE eigenvalue problem is closely related to the SPARSE matrix inversion and has been divided into three phases:

1. Changes in the subroutine set up.
2. Changes in the pivot selection strategy.
3. Changes in the subroutine SPARSE.

Changes in subroutine SETUP have been completed. The changed routine has yet to be tested in combination with other routines. Changes in pivot selection strategy and subroutine SPARSE are presently underway.

The subroutine SETUP has been changed to obtain the following configuration of matrix A and B. Where columns are concerned, matrices A and B appear to be lying side by side so as to give a (N x 2N) matrix.

* Ax = λBx and the Generalized Eigen Problem, G. Peters, and J. H. Wilkinson, SIAM Journal on Numerical Analysis, 7, #4, December 1970.

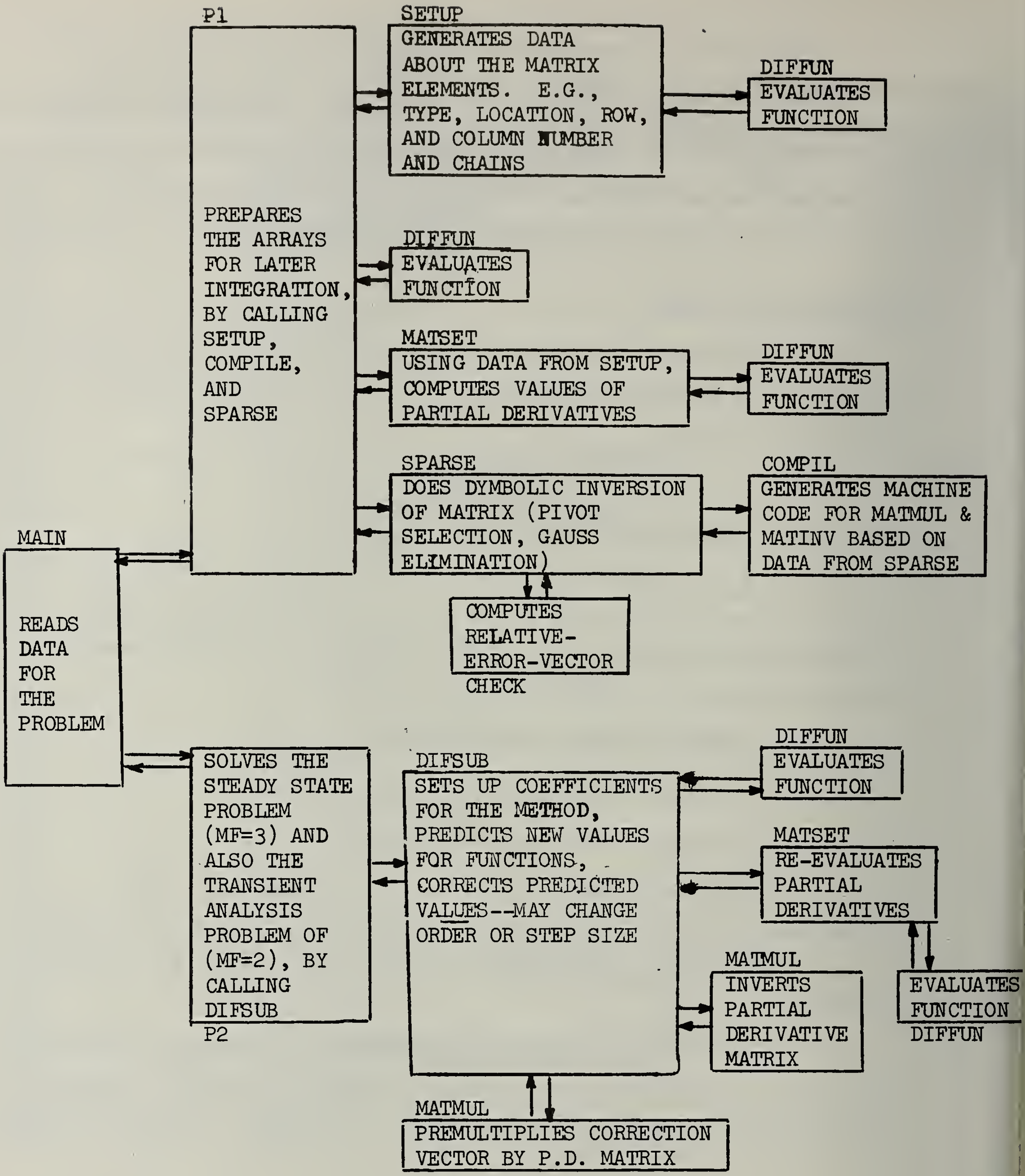

P1
PREPARES THE ARRAYS FOR LATER INTEGRATION, BY CALLING SETUP, COMPILE, AND SPARSE
SETUP
GENERATES DATA ABOUT THE MATRIX ELEMENTS. E.G., TYPE, LOCATION, ROW, AND COLUMN NUMBER AND CHAINS
DIFFUN
EVALUATES FUNCTION
DIFFUN
EVALUATES FUNCTION
MATSET
USING DATA FROM SETUP, COMPUTES VALUES OF PARTIAL DERIVATIVES
DIFFUN
EVALUATES FUNCTION
SPARSE
DOES DYMBOLIC INVERSION OF MATRIX (PIVOT SELECTION, GAUSS ELIMINATION)
COMPIL
GENERATES MACHINE CODE FOR MATMUL & MATINV BASED ON DATA FROM SPARSE
COMPUTES RELATIVE-ERROR-VECTOR CHECK
MAIN
READS DATA FOR THE PROBLEM
SOLVES THE STEADY STATE PROBLEM (MF=3) AND ALSO THE TRANSIENT ANALYSIS PROBLEM OF (MF=2), BY CALLING DIFSUB
P2
DIFSUB
SETS UP COEFFICIENTS FOR THE METHOD, PREDICTS NEW VALUES FOR FUNCTIONS, CORRECTS PREDICTED VALUES--MAY CHANGE ORDER OR STEP SIZE
DIFFUN
EVALUATES FUNCTION
MATSET
RE-EVALUATES PARTIAL DERIVATIVES
MATMUL
INVERTS PARTIAL DERIVATIVE MATRIX
EVALUATES FUNCTION
DIFFUN
MATMUL
PREMULTIPLIES CORRECTION VECTOR BY P.D. MATRIX

Figure 2

Where rows are concerned, the matrix A appears to be riding over B so as to give a (2N x N) matrix. The situation is shown in Figure 3. This configuration is considered convenient for handling by pivot selection strategies and the SPARSE program.

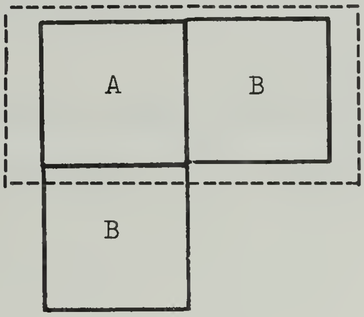


Figure 3

3.1.2 The Steady State Package (B. van Melle)

A new version of DIFSUB was prepared and tested this quarter employing a new startegy for solving the steady state problem (MF = 3). The integration is started as before, but only one pass is made through the corrector loop. After each step, the DY's are checked to see if they are converging

$$(D = \sum_{i=1}^{N} (DY(i))^2 \text{ is reduced}).$$

If so, certain parameters are changed to bring the process towards a Newton method. These parameters are H, the step size, which is made to approach the value of 1, and ALPHA, used in the corrector loop, which starts at 1 and moves toward 0. However, if D is not reduced, we restore ALPHA to 1 and reduce H in an attempt to reduce D.

Several strategies were also tested to minimize the number of function calls and matrix inversions. DIFFUN is called only once for each successful step, and the Jacobian is re-evaluated only when the previous step is unsuccessful (D was not reduced) and then only if it has not been recently re-evaluated. The problem then becomes one of moving toward a reasonable rate of convergence without "backing up" too often. The Newton method converges rather quickly when near the solution. But the rate at which we vary H and ALPHA to approach this method is important. On the simpler examples, the sooner we go to the Newton method, the faster the convergence, but in general it appears more efficient to proceed cautiously.

The new DIFSUB using these ideas was tested with several examples, and the results follow, with a comparison of the same examples run with the old DIFSUB.

Ex. 1 DY: $\sin(y_1 y_2) - y_2/2\pi - y_1$

$(1 - 1/4\pi)(\exp(2y_1) - e) + ey_2/\pi - 2ey_1$

Ex. 2 DY: $y_1^2 - y_2 + 1$

$y_1 - \cos(\pi y_2/2)$

Ex. 3 DY: $4 + y_1 + y_2 - y_1^2 + 2y_1\ y_2 + 3y_2^2$

$1 + 2y_1 - 3y_2 + y_1^2 + y_1 y_2 - 2y_2^2$

Ex. 4 DY: $y_1^2 + y_2^2 + y_3^2 - 5$

$y_1 + y_2 - 1$

$y_1 + y_3 - 3$

Ex. 5 DY: $\exp(y_1) + y_1 y_2 - 1$

$\sin(y_1 y_2) + y_1 + y_2 - 1$

Ex. 6 DY: $y_i' + G(i)(y_i - 1) - (y_i - 1)^2$, $i = 1, 2, 3, 4$

$y_5\ (G(1)T(3)/(T(3) + G(5)) - y_1) + y_6$

$2y_7 - y_7^3 - y_6 - T(2)$

$\ell_1 - y_7 - y_6 + 1$

$\ell_2 - \ell_1 - (y_1 + y_3 + y_4 - y_2)/2 + 1$

S1: $G(5) = 1 + G(1)$

$T(2) = \exp(-T(1))$

$T(3) = \exp(-G(1)T(1))$

Ex. 7 DY: $T(2) + {}_1 - 6y_3 - y_3'$

$y_3 - y_2'$

$y_2 - y_1'$

$\ell_3 + 11y_2$

$\ell_2 + 6y_1$

$\ell_1 - \ell_2 - \ell_3$

S2: $T(2) = \sin(T(1)/10)$

All examples used the initial values

$$y_i = 1 \ , \ i = 1, \ldots , N - NL$$

$${}_i = 1 \ , \ i = 1, \ldots , NL$$

and convergence was determined by the condition

$$\sum_{i=1}^{N} (DY(i))^2 < 10^{-6}$$

	OLD DIFSUB			NEW DIFSUB		
	NS*	NFNS	NW	NS	NFNS	NW
Ex. 1	130	678	120	116	149	10
Ex. 2	12	37	7	21	29	3
Ex. 3	24	66	8	21	32	4
Ex. 4	21	70	12	22	33	3
Ex. 5	13	41	8	19	27	3
Ex. 6	164	1739	162	19	42	3
Ex. 7	15	55	9	12	19	2

NS - number of integration steps

NFNS - number of function evaluations

NW - number of Jacobian evaluations/matrix inversions

3.1.3 Steady State Package Testing (W. Chung)

Graphics-User Interface Routine TRNØUT*

The subroutine TRNØUT, which is called in subroutine P2 for the transient analysis, is written and tested. It is composed of three parts which perform the following functions:

1. Communication with a user to obtain data
2. Integration by calling DIFSUB
3. Graphics plotting on the screen.

```
* SUBROUTINE TRNOUT(N,NL,T,G,Y,YL,SAVE,H,HMIN,HMAX,EPS,MF,YMAX,
 1     ERROR,KFLAG,JSTART,MAXDER,PSAVE,YLSV,F,DY,ERSV,F1,IP,IQ,
 1     DEL,TEND,YINI)
```

Data which is to be specified by a user for the purpose of plotting is as follows:

1. Variable names (Y and/or YL) to be plotted.
2. TEND maximum time limit.
3. IPNT number ofpoints where the integration and plotting are requested.

The program has been modified through three steps to provide more flexibility and sophisticated features such as error checking and multiple plotting. Simple descriptions are given below.

1. First version--simple and crude.
    a. only a single variable can be plotted.
    b. it calls the subroutine pair MESAGE/REPLY to communicate with a user at a graphics terminal. User-provided data is recognized and converted to binary by a FINDNM/NUMCHR pair. However, since subroutine FINDNM handles only fixed point numbers, the above procedure should be repeated for floating point numbers, which is considered very inconvenient.
    c. Plain axis and curve are drawn without any scaling and identification.
2. Intermediate version.
    a. multi-variable plotting in a very restricted manner.
    b. use of unit 1 and 2 for graphics I/Ø--no problem for floating point numbers.
    c. to compute the function value at an exact point of time, Y variables are interpolated from the derivatives. Since we cannot interpolate YL variables, points where the actual plotting is done are saved in an array TSV(I).
    d. user is allowed to change parameters and start again.
3. Present version.
    a. flexible multi-variable plotting, which allows any Y and/or YL variables (total number is limited to 15

because of visibility on the screen) .to be plotted in any order at a time.

Names of Y and YL are stored in arrays IYN and IYL, respectively, and the storage used is optimized by saving the integrated values of Y and YL at plotting points in a two-dimensional array YSV(I, J) such that Y(1, IYN(I)) are stored in YSV(I, J) and YL(IYL(I)) in YSV(I+NY, J) where NY is the number of Y variables to be plotted.

b. interpolation of Y is eliminated not only because time spent in computation is not worthwhile but also because it is incompatible with YL variables if both Y and YL are plotted using the same time axis.

c. error checking and recovery features are provided; i.e., each datum which is specified through the terminal is checked for errors, and branching is taken to various points in the program according to the error conditions.

d. axes are scaled to aid reading. Each variable is identified by putting a function name at the tail of each curve.

Communication with a user (question-answering) is implemented by using two kinds of I/Ø sequences.

1. WRITE(2,*)/READ(1,*) pair.
This is used mainly to get numeric data.

2. CALL MESAGE/CALL REPLY pair.
This is used for non-numeric data.

Test of TRNØUT on batch was done only for the numerical portion of the program because conflicts between units 1/5 and units 2/5 could not be resolved at the time.

Modification of P2

Due to the implementation of TRNØUT, a subroutine P2 now calls TRNØUT instead of DIFSUB. Also, initial values of Y and YL are saved in an array YINI(40) immediately before P2 calls TRNØUT. These saved values are used for initializing Y and YL everytime DIFSUB is called in TRNØUT.

3.2 Non-Numerical Packages

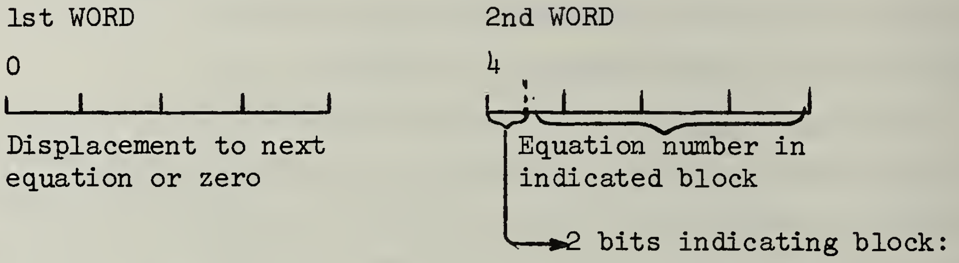
3.2.1 Item Analysis (J. Koch)

Item analysis was modified this quarter to allow it to be run from batch or interactively with the PDP-8. New commands and error messages were added and the section to create the node/connection table was re-written. The equation format was changed by the addition of an extra word to identify the equation to the user.

Using the graphics filing system, item analysis is able to retrieve any element, network, or mnemonic the user has previously sent to the 360 (see command list). When a picture is sent from the PDP-8, it is immediately stored in PICLIB; then item analysis creates and places the coded output data structure onto ITEMLIB. For PICLIB, the key is created from the six character picture name followed by two bytes that indicate which block of the picture it is (i.e., '0800' for header, '0801' for local lines, '0802' for comment text, 0803' for subpictures, '0805' for node connections and terminals, '0806' for declarations, '0807' for equations and parameters). Itemlib also uses the six character picture name, but appends two EBCDIC blanks ('4040') to the key. Also, any node type declarations that are discovered are written out on NODLIB with their eight character name as a key. The fourth file used is ERRLIB which contains all the syntax errors detected by item analysis. These errors are sent to the PDP-8 to be displayed on the screen and are also written on the line printer for hard-copy reference. At present, error messages appear with no specific reference to the line that was being examined at the time the error was noted. This may be remedied by outputting the error message followed by the line it references. ERRLIB can also be destroyed by the user after he has noted all the errors. The key for the error message is created by the filing system through the use of the XAPPEND command.

The names of these four files were modified so that they are each preceded by the eight character log-on name of each user. Otherwise, every user would be working from the same files and could modify or delete the work of other users. (At present, all test elements are in POOHBEARPICLIB, etc., to allow them to be referenced when item analysis is run from batch). Upon entry to item analysis, the address R1 + 12 is the eight character user log-on. This name is then moved in front of the four file names before any commands are processed.

An extra word was put in each equation following the first word.


1st WORD
0
Displacement to next equation or zero
2nd WORD
4
Equation number in indicated block
2 bits indicating block:

00--element equation
01--network equation
resolved in Global Analysis

10--default equation
11--parameter equation
resolved in Item Analysis

PARSE is the subroutine that parses all the equations in an element. Some new parameters and bit indications were added. The complete parameter list is:

```
PFLG      DS    X                  POSSIBLE FLAG BITS:
VLHBIT    EQU   X'80'              ONLY 1 VARIABLE ON LEFT SIDE OF =.
HALTBIT   EQU   X'40'              ON IF PARSE HALTED.
ERRBIT    EQU   X'20'              ON IF ERRORS DETECTED BY PARSE.
PADDR     DS    AL3                ADDR OF WHAT I AM TO SCAN.
PEND      DS    A                  END OF WHAT TO SCAN.
VLOC      DS    A                  LOC OF LOCAL VARIABLES.
VGLB      DS    A                  GLOBAL.
VPAR      DS    A                  PARAMETERS.
VEI       DS    A                  E/I NAMES.
VLH       DS    A                  LH SIDE VARIABLES.
VCON      DS    A                  CONSTANTS.
FRES      DS    A                  WHERE TO GET FREE SPACE.
LASTEQN   DS    A                  ZERO WHEN STARTING NEW EQN TREES.
COUNT     DS    A                  EQN NUMBER IN BLOCK.
S8WORK    DS    A                  ADDR OF WORK AREA FOR MESS. TO PDP8.
ERRLIST   DS    A                  ADDR OF ERRLIB DCB FOR PARSE ERRORS.
TRANS     DS    A                  ADDR OF EBCDIC TO ASCII TRANS. TABLE.
FILE      DS    A                  ADDR USED BY XAPPEND IN PARSE.
```

PARSE can now handle any number of expressible in double precision floating point notation. Also, exponents of 10 can be indicated through the use of either a D or an E (i.e. 0.173768D-10, .987E30}.

Command List

(ALPHA is a valid six character element or network name)

1. 'ITEMIZE ALPHA'--causes ALPHA to be read from PICLIB and put through item analysis.

2. 'ANALYZE ALPHA'--causes item analysis to call global analysis with the parameter ALPHA. Item analysis does no processing of ALPHA.

3. 'DESTROY ERRLIB'--causes item analysis to delete the user's error library. This command is only valid for destroying the error library and will not be recognized if any other file name but ERRLIB is used.

Messages to User

1. 'SYNTAX ERRORS DETECTED.'

   If errors detected by item analysis.

2. 'UNKNOWN COMMAND.'

   An illegal command was received.

3. 'INVALID BLOCK TYPE RECEIVED.'

   An error in transmission occurred.

4. 'ITEM READY FOR FIRST INPUT'

   Indicates item analysis is loaded and ready.

5. 'ITEM ANALYSIS COMPLETE'

   The analysis of an element is done and item analysis is ready for another input.

6. 'RECORD XXXXXX-XXXX NOT FOUND. ENTER AGAIN.'

   When an 'ITEMIZE' command is given and the indicated block does not exit on PICLIB.

7. 'MNEMONIC STORED.'

   After user has sent a mnemonic.

8. 'ERRLIB GONE...'

   After the 'DESTROY ERRLIB' command is completed.

Error Messages

1. 'NO : IN NODE TYPE DECLARATION. NAME: E(X,Y) I(P,T9)'

   i.e., 'ELECTRIC E(A,B), I(C,D)'

2. 'NO) IN NODE TYPE DECLARATION. NAME: E(X), I(A,R)'

   i.e., 'ELECTRIC : E(A,B), I(X)'

3. 'VARIABLES IN NODE TYPE DECLARATION NOT E OR I.'

   i.e. 'ELECTRIC : W(A,B,C), Z(Q,R,T)'

4. 'VARIABLE REDEFINED.'

   Variable defined twice as either a global, local, or parameter.

5. 'INVALID ENDING CHAR IN PARAMETER DEFINITION.'
Must end with a single variable or an equation.

6. 'NO E OR I VARIABLES SPECIFIED FOR TYPE.'
Just a type with no variables was entered.
(i.e., 'ELECTRIC :')

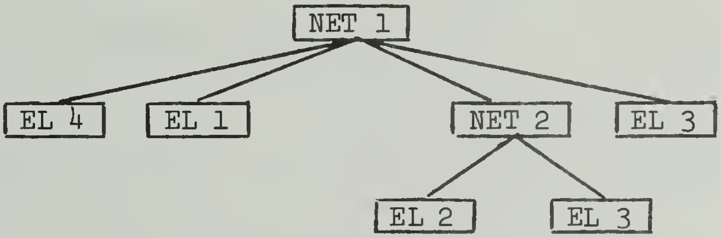
7. 'NON-EXISTANT TERMINAL TYPE REFERENCES.'
When a terminal has been assigned a type but the type was later deleted from the list of types.

8. '******* ILLEGAL XXXXXX DETECTED'
Message originates in PARSE and XXXXXX refers to the level of compilation at which the error was detected in an equation.

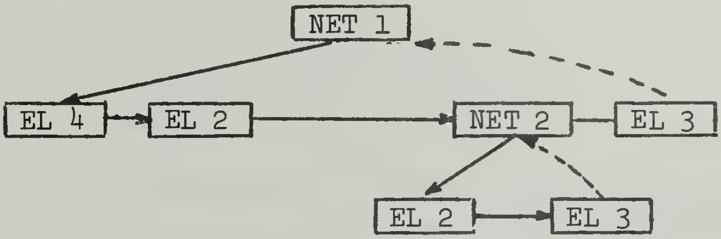
3.2.2 Global Analysis I and II (S. Wilkins)

When errors occur during the numerical analysis of a network, symbolic information known to the user must still be available in order to provide valuable diagnostics. All tables created by Global Analysis II have been expanded to facilitate the diagnostic process. During Global II a tree is constructed reflecting all the networks and elements in the structure of the outer network and their interrelationship.

If a network had the following structure



Then the tree constructed would reflect this structure as:



The complete description of the linkage of this tree structure was given in an earlier report.

Each entry in the tree structure is 30 bytes containing the following information.

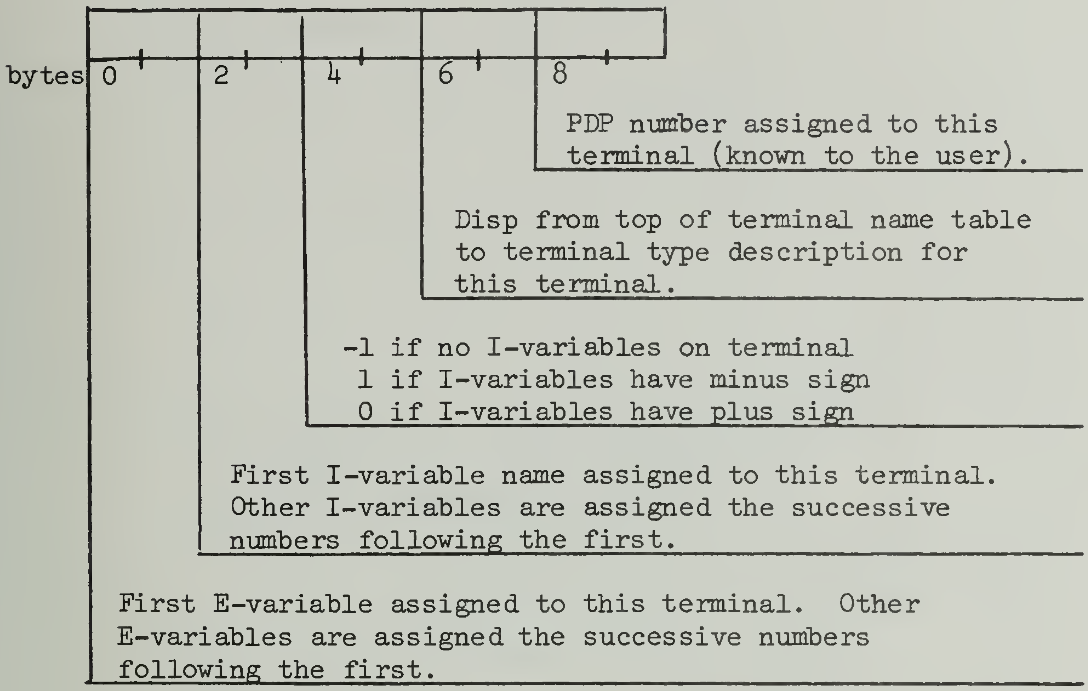
BYTES	TYPE	DESCRIPTION
0-1	Numeric	Disp* to 1st element in subpicture if this is a network; zero otherwise.
2-3	Numeric	Number of local variables associated with this element
4-5	Numeric	Disp from top of local variable list to local variables associated with element.
6-7	Numeric	Disp to next element on same subpicture level or if bit 0 = 1 then bits 1-15 are disp to network of which this element is a subpicture.
8-9	Numeric	Displacement to Listone associated with this element.
10-11	Numeric	Number of entries in Listone.
12-13	Numeric	Subpicture level of this element counting from level 1 which is associated with main (outer) network.
14-15	Numeric	≠Number of global variables associated with this element.
16-17	Numeric	≠Disp from top of global variable list to the variables associated with element.
18-10	Numeric	Number of parameters associated with element.
20-21	Numeric	Disp from top of parameter list to parameters associated with this element.
22-29	Alpha	Symbolic name of element.

*All displacements unless otherwise explained are calculated from the top of the internal tree.

≠Not always used since some Global Variables are placed in the internal variable list rather than the global variable list.

Listone is a table for every element occurring in the structure of the network giving the E and I variable names assigned to each terminal of the element.

Each entry (one per terminal) has been expanded to 10 bytes.



PDP number assigned to this terminal (known to the user).

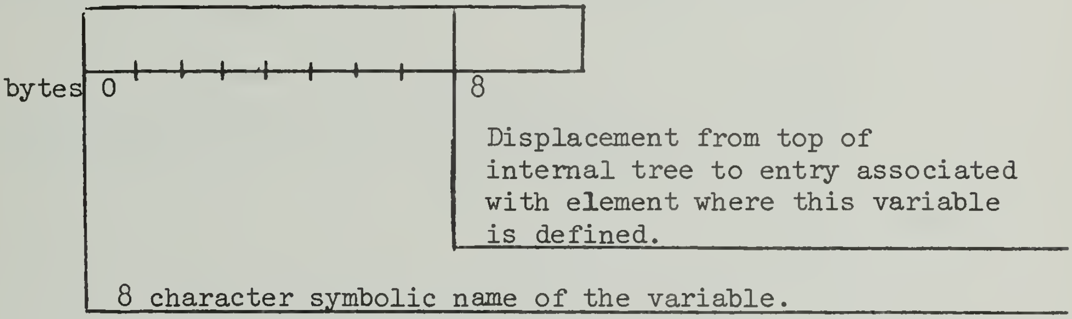
Disp from top of terminal name table to terminal type description for this terminal.

-1 if no I-variables on terminal
1 if I-variables have minus sign
0 if I-variables have plus sign

First I-variable name assigned to this terminal. Other I-variables are assigned the successive numbers following the first.

First E-variable assigned to this terminal. Other E-variables are assigned the successive numbers following the first.

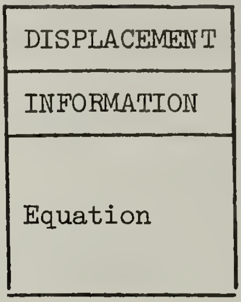
The three variable tables have been expanded to 10 bytes per entry.



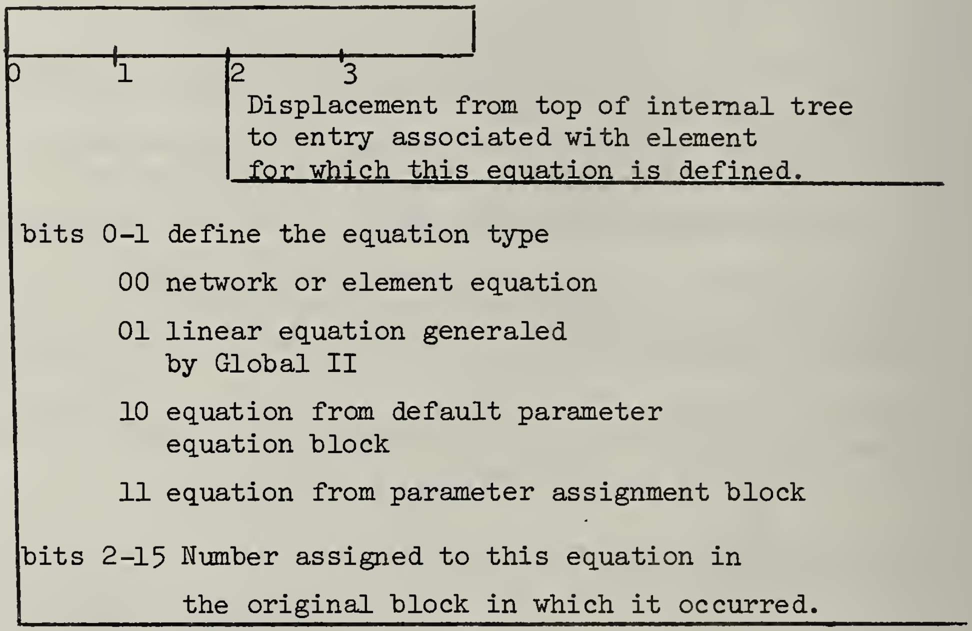
Displacement from top of internal tree to entry associated with element where this variable is defined.

8 character symbolic name of the variable.

The beginning of each equation in the output equation block has been expanded to include an extra four bytes of information



Information bytes



Displacement from top of internal tree to entry associated with element for which this equation is defined.

bits 0-1 define the equation type

00 network or element equation

01 linear equation generaled by Global II

10 equation from default parameter equation block

11 equation from parameter assignment block

bits 2-15 Number assigned to this equation in the original block in which it occurred.

3.3 Graphical Remote Access Support System (GRASS)

3.3.1 Disk Monitor System (M. Michel)

Production use of the DMS continued this quarter without the occurrence of any fatal errors. (c.f. section 3.3.4)

3.3.2 Display Terminals (R. Haskin)

The display terminal control program, ACID, has been used extensively this quarter, and has survived with only minor modification.

A slight improvement was made to the dynamic buffer deallocator which enabled it to release buffers without blowing up.

Heavy multiterminal operation revealed that hardware noise problems (very) intermittently cause data errors and consequent loss or garbling of display files. Several software modifications were implemented to lessen the effect of this problem.

Another minor hardware problem noticed was the again very intermittent failure of the terminal to return coordinates after a read cursor operation (joystick hit). Hitting any key on the terminal keyboard twice will recover from the problem.

All in all, the performance of both ACID and the Computek terminals has been satisfactory, and this part of the system can now be considered as fully operational.

3.3.3 Information Retrieval Package (M. Michel)

Use of the package became more regular and heavy this quarter. Response remained good as the number of files in the directory increased. Resorting of the block list during an operation was extensively tested and shown reliable. Two errors were found and corrected. Saving a file with the overlay option, if the file named already existed, resulted in improper deletion of the blocks of the existing file. This caused some free blocks to be lost from the unsorted free-list. Also, the pointers controlling the unsorted free-list were not being saved and restored by GLØAD/GEXIT.

The version of the package sent to the Griffiss Air Force Base, Rome, New York, has been modified by them, quite easily, to run with a DEC RF08 disk. Debugging of their version is just entering its initial stages, but the outlook for bringing the system up rapidly is quite good.

3.3.4 Disk Interface (C. Hyde)

The disk and associated hardware miraculously survived two floods and numerous power-downs (all due to malfunctioning air conditioning equipment). Only a few minor track failures occurred. An occasional, though devastating, error that has appeared but eluded discovery, is the loss of the first two words of a record on a write operation. A new test program was written, but failed to pinpoint the source of the error.

3.3.5 Monitors (M. Michel)

GRASP (360 Remote Monitor)

A perfect performance record was scored this quarter; no abnormal termination occurred and all functions operated properly and efficiently. Some additions were made to increase error detection of bad 2701 records; namely, extra and more complete log messages. The only contemplated alterations remaining are procedures for performing accounting operations, such as counting subtask I/Ø, attaches, logon/logouts, etc.

2701 Data Link

Adequate performance has continued on the data link. No serious problems have occurred.

GLASP (PDP-8 Local Monitor)

Extensive file-menuing ability was added this quarter. Saving or deleting a picture or mnemonic causes the appropriate function to be performed in the list of file names maintained in a special menu file for each user. The alphabetical list can be displayed in segments of 32 lines each during the operation of the generalized drawing package (GNDRW). Indicating an origin in the draw area and then "picking" a displayed mnemonic with the joystick causes the desired mnemonic to be integrated into the current picture data structure and to be appropriately displayed at the origin. At logout, the new menu file is saved in I.R.; the next time the same user logs on, his new menu will be displayed.

An option was added to CALLER due to the presence of the menu feature: purgeout. This causes all files named in the menu file to be deleted locally, then a regular logout is performed. Hence, a user can easily free the library space he used to make it available to others currently on the system.

The time of day has also been added to the upper-righthand corner of the display area. Whenever the display is regenerated, the updated time is also added in the form HH:MM:SS (hours, minutes, seconds).

Having the time available has enabled the addition of still another feature to CALLER: a system log. Each time a user logs on, logs out, or unsuccessfully attempts to log on ('threat'), the time, terminal number, event, and user ID (or logon string in the event of a 'threat') is output to the TTY printer.

Production sessions for two and three terminals have been held and were quite successful. Response time was adequate at each terminal, even for the worst case of three simultaneous picture regenerations. In addition, extensive testing and debugging of the lockout feature of the monitor (and associated DSMN routine) was performed. Two and three terminals were driven from a total of six core pages (as opposed to the usually available thirty-two) for small, medium, and high complexity pictures. All lockout situations were properly caught.

GLASP Status

The monitor with all basic features currently available is quite stable and well debugged. Additional debugging and "tuning" is being performed as increased activity turns up minor problems. Also, user feedback is beginning to be felt. For example, it was found that the original position of the 'purgeout' box caused some not-so-small annoyances. 'Purgeout' is in the same position as 'Return' in most other modes. Hence, force of habit led to an undesired menu (and file!) deletion. This is being changed, naturally.

Additional features are being planned and implemented. The basic capabilities of the system, while adequate for some activities, are hardly a complete or sufficient set. Increased documentation is also now in the pipeline.

Applications Support

Interactive development of the Simulation and Modeling System under GRASS has continued this quarter.

Extensive use has been made of the batch debugging program.

Use of the system for investigating methods for stiff equations has begun.

3.3.6 Remote Data Structure Utilities (R. Haskin, J. Nickolls)

Data Structure Communications Package

The communications and data structure handling program COMMUNE has been thoroughly tested this quarter, and is now being used by other programs in the system. The routines have been thoroughly documented, and a write-up is now available. Several important improvements and extensions have been made to the package, with care taken to preserve compatibility with earlier versions. These changes are:

CALLER: The FORTRAN link to the package has been completely rewritten. It now supports the FORTRAN I/O package (F1OCS#.modification) and includes a much more efficient linkage to routines in COMMUNE.

LINE, LINED, TEXT: X and Y coordinates may now be specified either in screen increments (as before) or in floating point inches. CALL LINED (0., 1., 1) will now cause a one inch vertical line to be added to the line block. Also, an optional error return is provided for these routines in the event that the data structure block being built fills up. For example, the FORTRAN statement CALL LINE (10, 30, 1, & 400) will transfer control to statement 400 if the line block becomes full as a result of the call.

REPLY: An optional fourth parameter has been added in which, if present, will be returned the actual number of characters sent by the PDP-8 before being padded with blanks. The calling sequence for this option is:

CALL REPLY (ITYPE, LEN, ITEXT, NCHARS)

The old calling sequence can still be used if NCHARS is not desired.

MESAGE: This routine sends a text line to the display. The calling sequence is:

CALL MESAGE (ITEXT, LEN)

ITEXT is either an INTEGER*4 array containing the characters to be sent or is a character constant. LEN is the number of characters to be sent ($\leq$ 72)

SEND: An option is available which will cause the line block to be sent to the display and re-initialized if it fills up before another SEND call. To use this option, the block must not be marked to replace the current PDP-8 data structure, and must be for display. To do this:

```
ICODE(2) = 0
ICODE(3) = 0
SEND(0, ICODE)
   .
   .
   .
   calls to LINE and LINED, possibly
   filling up several line blocks
   .
   .
   .
SEND(1, ICODE)
```

This option behaves as follows:

(a) it remains in effect until the line block is sent explicitly (as by the second SEND call above),

(b) if more than one block is sent, only the first will cause an erase or frame,

(c) when the line block is finally sent explicitly, the ICODE parameters from the SEND(0, ICODE) are used (erase and frame behave according to (b),

(d) sending the line block explicitly clears the option.

Hardcopy Output

This quarter the routines to produce hardcopy CalComp plots from pictures drawn on a console were incorporated into a package called GPLØT.

The input to GPLØT consists of the name of a picture or mnemonic stored in the graphics filing system, and a string of option words specifying what parts of the picture are to be drawn. This input may be specified via a PLØT command from a console using the library service package, or from data cards. A PLØT command such as:

"PLØT_PICTURENAME/USERNAME.LIBRARYNAME_OPT1,OPT2,OPT3"

is stored as an entry in the file SYSTEM.PLØTLIB for later execution by GPLØT, which plots all such entries and those from cards as a batch job.

Picture names entered on data cards must be fully qualified names containing username and library name, but these may be omitted when using the PLØT command. LSD automatically adds the logon username 'PICLIB' in this latter case if not already present in the command.

The options currently available are listed below. Each option word is five characters long and begins with a one letter prefix describing what it affects. Any option may be preceded by a negation symbol '¬' to indicate the negative sense of the option. Starred options are defaulted to ØN.

P-options concern the entire picture:

*PLLIN - Draw local lines from line block
*PCTXT - Print comment text from text block
*PSUBP - Draw subpictures from mnemonic instance block
*PTERM - Draw terminals and connections from terminal block
PDECL - Print declarations
PEQPM - Print equations and parameters
*PFRAM - Draw frame as seen on console
*PFTXT - Print system data in the frame

M-options concern mnemonic storage blocks:

*MLLIN - Draw local lines of mnemonics
MCTXT - Print comment text of mnemonics
MTERM - Draw terminals (when only drawing the mnemonic)
*MPARM - Print model parameter list beside mnemonic
MTNM# - Print type name number on mnemonic terminals
MTNML - Print mnemonic type name list in menu area

T-options concern terminal blocks:

TSEQ# - Print sequence number on terminals

TTNM# - Print type name number on terminals

TTNML - Print type name list in menu area

TDINV - Draw "invisible" connections between terminals

S-options concern subpicture instance blocks:

SPARM - Print parameter assignments of subpicture instances

The user may specify as many options on as many cards or console lines as needed by ending the line with a comma and continuing on the next one.

When GPLØT is run from batch, it first plots all jobs specified in data cards, then starts reading jobs from SYSTEM.PLØTLIB. All routines accessing SYSTEM.PLØTLIB use the system macro ENQ to ensure that no plot jobs are lost. GPLØT deletes all entries from SYSTEM.PLOTLIB when it is finished.

If desired, a mnemonic may be plotted by itself by using the six character mnemonic name followed by a period and the mnemonic number:

"PLØT_TRNSIS.1_MCTXT,MTERM,MTNML".

Graphics Library Maintenance Program

LSD has been expanded to allow common access of files among users. This has been done by expanding the command syntax to provide for the qualification of all file names (i.e. picture and mnemonic names) with the library from which they are to be taken. Also, library directed commands, such as CATALOG, allowed specification of the library to be used. The new syntax is:

Generalized file name--

$$\left\{\begin{array}{l}\text{<picture name>}\\ \text{<mnemonic name>.<mnemonicid\#>}\end{array}\right\}\left[\text{/[<username>.]<libraryname>}\right]$$

Examples--

A. PLOT RESIST

B. PLOT RESIST.Ø

C. PLOT RESIST.Ø/PICLIB

D. PLOT RESIST.Ø/HASKIN.PICLIB

E. CØPY RESIST.1/HASKIN.PICLIB NEW.0/NICKØLLS.ALTLIB

Generalized library name--

[[<username>.]<libraryname>]

Examples--

E. CATALOG

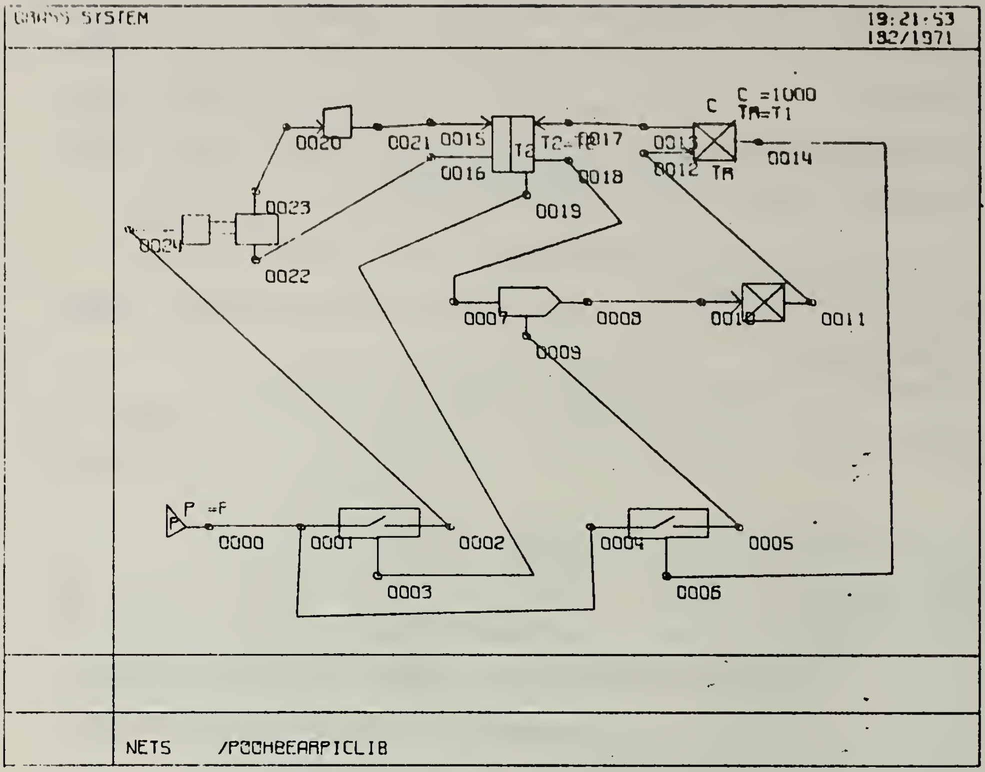
F. CATALOG PICLIB

G. CATALOG HASKIN.ERRLIB

The defaults for specifying a library name are as follows:

1. If the username is omitted, the name used is the one by which the user is logged onto the 360. (i.e. examples C, F above).

2. If the entire library name is omitted, the username is chosen as above, and the default library name is PICLIB (i.e. examples A, B, and E above).

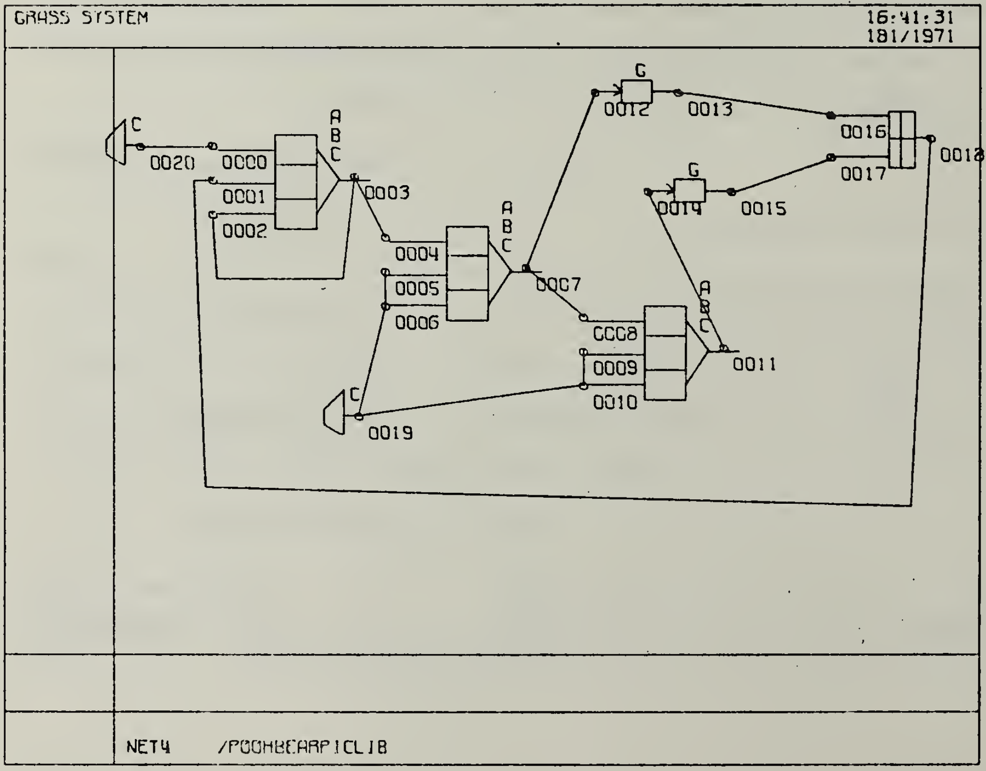


EQUATIONS:

F=MIN(VS,TIME*V

PARAMETERS:

T1
T2
VS



DEFINITIONS:

D

EQUATIONS:

D=SIN(TIME/10)

SAMPLE HARDCOPY PRINTS OF TWO MODELING SYSTEM NETWORKS

3.4 Computer Maintenance and Construction (C. Carter, I. Cunningham, C. Hyde, T. Kerkering, H. Lopeman, B. Miller, J. Nickolls)

3.4.1 Computek Computer Graphics Terminal

STATUS:

The three University Purchase Orders (#215711, #228151, and #233131) are complete and the items are presently out of warranty.

The remaining items promised to us but not delivered are:

1. Modifications necessary on 611 storage scope.
2. Graphic allignment section.
3. Software levels 0, 1, 2.
4. Fix for cursor intensity during write-through.

SPARES:

Replacement spares have been checked out on the two Computek terminals and a third terminal has been assembled incorporating the spare keyboard and storage tube. This third terminal is operational and a part of the PDP-8/I multiplexor graphic terminals.

3.4.2 Equipment Maintenance Log Summary

Computek

1. Terminal #1 power supply +150 regulator blown. (Repaired)
2. Terminal multiplexor losing flags and going to initial state. (Logic error in multiplexor)

PDP-8/I

1. Erratic problem of occasional channel hang-up. (Replacing filter capacitors with larger ones on channel boards)
2. Indicator lamps out. (Replaced)

PDP-8

1. Core memory bank 2 failing constantly. (Bad memory address selector board replaced)

2. Switches failing on PDP-8. (Replaced with new switches of different manufacture)

3. Top dectape drive slow to respond occasionally. (No remedy as yet)

Disk

1. Track 26_8 failures. (Rewired track)

2. Track 46_8 failures. (Rewired track to spare)

3. Track 128_8 failure. (Rewired track to spare)

Inktronic Printer

1. Nozzle clogged in column 2. (Cleaned)

2. Paper drive motor clutch frozen, burned out motor drive board resistor. (Repaired clutch and board)

3.4.3 Air Conditioning (Room 31A)

There has been a rash of problems with the computer room air conditioning.

1. Control valve air compressors both failed causing heat and humidity build-up. (Thermal shutdown tripped)

2. Control stuck causing excessive heat in computer room. (Thermal shutdown tripped)

3. Condensate pan in air handling unit drain plugged up and spilled water through the ceiling into the computer room (twice). (Minor damage to some tools involved)

During this quarter, a thermal overload protector was installed to dump all power in the computer room if the temperature reached 85°F. It appears to work very well.

3.4.4 Stereomatrix Interface (I. Cunningham)

Construction suggested in the previous quarter has been brought close to completion. Installation and checkout is beginning. Specific work done:

Cards

E6/D6	A/D Data Buffer
E7/D7	DMA Control--Interrupt Flags
#8/D8	Direction Decode
E9/F9	Sequencer
E10/F10	Up Down Counter
E11/F11	Up Down Counter
E12/F12	Up Down Counter
F16	Line Receiver
F15	Line Drivers
F14	Line Drivers
F13	Line Drivers
C19/D19	IOT Decoder

Cables

In addition, six cables were made and checked out. Main frame wiring completed from the IOT decoder to the pallet cards connector block.

Five twisted pair cables and three coaxial cables have been strung from Room 31E for about 150 feet to Room 28 in the basement where information will be transmitted and received.

Mainframe

A rack containing the A/D converters and power supply was wired. A DEC rack was wired which contains all of the above mentioned cards except for the IOT decoder.

Construction of the interface between the PDP-8/I and the stereomatrix display has been completed and approximately half of it has been checked out. An additional power supply was added to the PDP-8/I to handle the extra logic and one bad card was found in the DMØ1 direct memory access multiplexor. The cables to the display have been installed but are not connected.

The total interface can be checked out without the stereomatrix display and this should be completed shortly.

Stereomatrix Interface Instructions

6511: load X beam coordinate position from AC(0-9) (10 bit sign and magnitude)

6516: read X cursor coordinate position into AC(0-9) (10 bit 2's complement)

652X: for Y positioning

653X: for Z positioning

6541: skip on line done flag

6542: stop display and clear line done flag

6544: load the 12 low order bits of the line address and start the display

6551: skip on OR of CURSOR, INCIDENT, and FUNCTION KEYBOARD FLAGS

6552: or CURSOR, INCIDENT, and FUNCTION KEYBOARD FLAGS into AC bits 0, 1, 2, respectively

6554: load extended address from AC bits 9, 10, 11 and clear CURSOR, INCIDENT and FUNCTION KEYBOARD FLAGS from AC bits 0, 1, 2, respectively, if that bit is a one

6561: or FUNCTION BOX bits into AC and clear FUNCTION BOX

6562: or current line address into AC

6564: resume drawing line at current point after incident interrupt

Line Elements

The format for line elements is as follows

```
0   1          5 6      11
| I |  DIRECTION | LENGTH |
```

WORD IN MEMORY

Each element can draw one straight line. If I is a 1 the line is intensified. Bits 1 through 5 permit the selection of 1 of 27 directions. The direction is calculated as follows:

DIRECTION = (ZD*3 + YD)*3 + XD

where XD, YD, and ZD are the values of X, Y, and Z unit vectors (0 is no component, 1 is a unit vector in the positive direction, 2 is a unit vector in the negative direction).

LENGTH is an unsigned integer with a maximum value of 63. If the DIRECTION is 36 octal the LENGTH is interpreted as status. Bits 10 and 11 specify intensity; bits 7 and 6, scale; and bits 8 and 9 are free. A value of 37 octal for the DIRECTION cause the display to halt and the line done flag to be set.

A picture is generated by first loading the X, Y, and Z registers to position the beam. The address of a sequence of line elements is loaded. The interface generates the points for the display until a HALT is reached.

The following program draws a cube.

```
                10F
LOOP            CLA
                6554            /LOAD EXT ADDRESS
                TAD         XCOORD
                6511            /LOAD X-CORD
                CLA
                TAD         YCOORD
                6521            /LOAD Y-COORD
                CLA
                TAD         ZCOORD
                6531            /LOAD Z-COORD
                 .
                 .
                 .
```

```
                    .
                    .
                    .
                    CLA
                    TAD          LINADR
                    6544            /LOAD LINE ADRS AND START
                    6541            /SKIP ON FLAG
                    JMP          .-1
                    6542            /CLEAR FLAG
                    JMP          LOOP /DRAW AGAIN
XCOORD              4200            /-40 OCTAL
YCOORD              0200            /+40
ZCOORD              4200            /-40
LINADR              36              /SET SCALE AND INTENSITY
                    4677
                    4177
                    4377
                    5177
                    4677
                    4277
                    4377
                    5277
                    4177
                    0677
                    5177
                    0377
                    4277
                    0677
                    5277
                    3700            /HALT
```

Form AEC–427
(6/68)
AECM 3201

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY–TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO. COO-1469-0193	2. TITLE 2nd QUARTERLY PROGRESS REPORT 1971 (April, May, June)

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report
□ b. Conference paper not to be published in a journal:
Title of conference ____
Date of conference ____
Exact location of conference ____
Sponsoring organization ____
□ c. Other (Specify) ____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.
□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

C. W. Gear, Professor
and Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature Charles W Gear.	Date June 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.
□ b. Report has been sent to responsible AEC patent group for clearance.
□ c. Patent clearance not required.

4. IMAGE PROCESSING AND PATTERN RECOGNITION RESEARCH: ILLIAC III (Supported in part by Contract AT(11-1)-2118 with the U.S. Atomic Engery Commission)

A balanced program of experiments with an image processing computer has been carried out. Highlights of the program include:

1) Extensions of the Show-and-Tell package for interactive picture processing.

2) Development of the theoretical basis (signal detection theory, covering techniques, etc.) for plane parallel image processing - procedures critical for rapid preprocessing and local feature extraction.

3) Strategies for structural inference using multigraph scene representation; undoubtedly techniques here will have utility far beyond their immediate application to scene analysis.

4) Applications have been initiated in diverse areas: to materials science, bio-medicine, robotics and remote sensing of the environment.

5) Finally necessary completions and development of the Illiac III Image Processing Computer integral to the above research are recommended.

Our goal here is to design a sight sensory system predicated upon parallel strategies for image processing. This development must of course evolve from our experience with the Illiac III system. The key property we seek is the ability of the system to learn rapidly from instructional interaction with professional personnel (not necessarily trained in computer technology). The validity of this orientation has been, we believe, vindicated by the rapid growth in significant image processing applications we can now attack at the end of the first year of this contract.

4.1 INTERACTIVE PICTURE PROCESSING

One objective of our research has been to develop a strategy which would permit personnel (not trained in computer technology) to input scanning instructions directly to the image processing system. An interim stage in this development has been the design and implementation of an interactive system for picture processing: The Show-and-Tell system described below.

4.1.1 Show-and-Tell

The Show-and-Tell programming system is a console-oriented software package providing a facile method of using the operational components of the Illiac III computer. The system presently operates on the hardware depicted in figure 1. Overall documentation of the Show-and-Tell system is contained in reference [1].

Show-and-Tell is designed to support local image processing and image acquisition directly using the Illiac III PAU and S-M-V Systems, and also to support experimentation in image processing theory using the high-level language and mass storage of the IBM 360/75. To this end, a bidirectional link with the IBM 360 is provided allowing (1) calling IBM 360 programs and subroutines from the PDP8/e console teletype and (2) transmission of data and pictures back and forth between Illiac III and the IBM 360 via a dedicated PDP8/e interface.

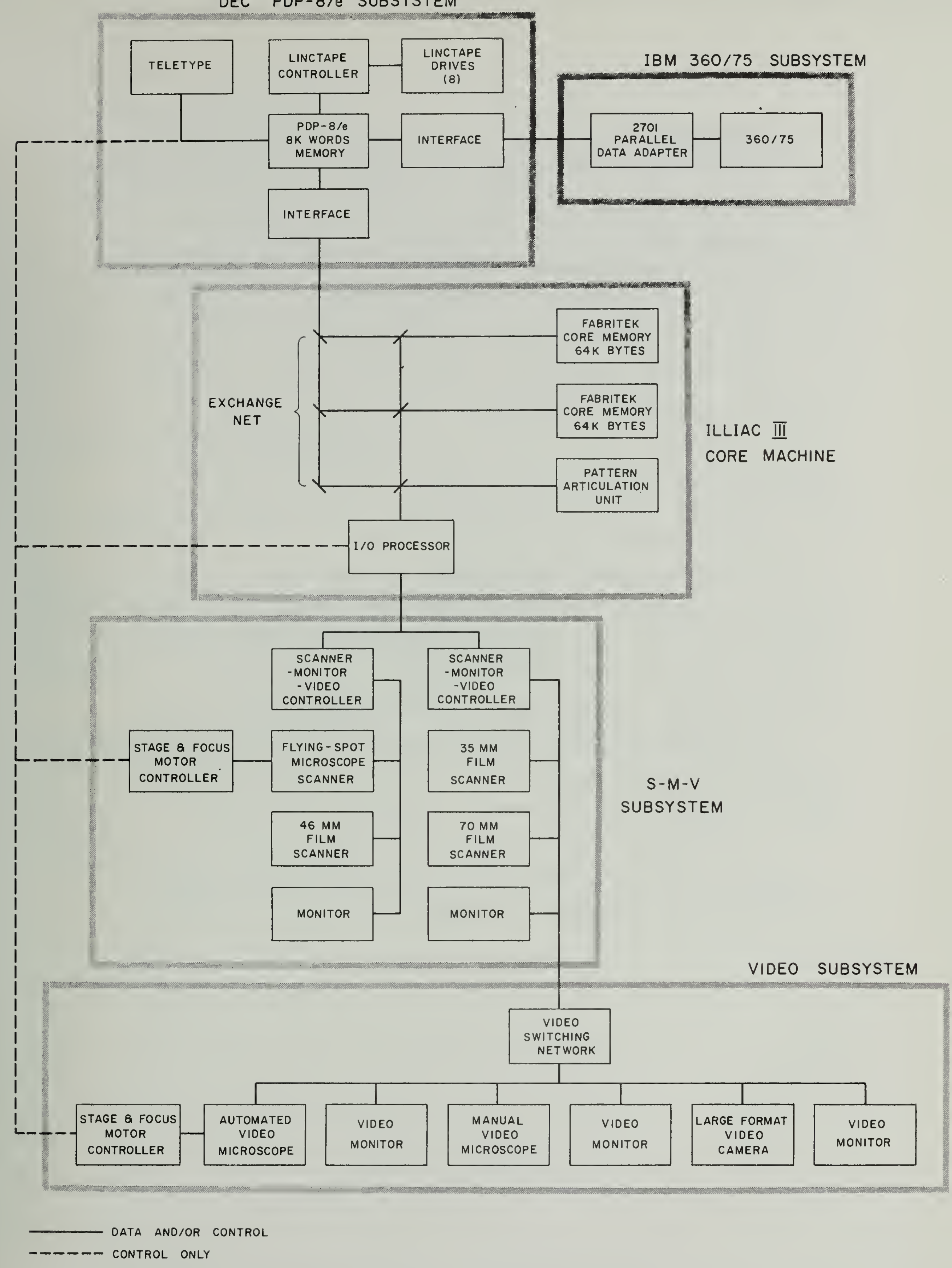


FIGURE 1. HARDWARE CONFIGURATION

4.1.2 Intermachine Link

The intermachine link provides a means for programmers to develop image processing codes using simulators for various sections of Illiac III (e.g. PAX II) until these sections are checked out. In this regard a Digital Equipment Corporation PDP8/e was ordered in January 1971. The machine will initially be used to communicate with the Department of Computer Science's 360/75. It will eventually assume its major functions as maintenance processor and data concentrator for Illiac III as the system evolves.

The PDP8/e was installed and accepted on May 26, 1971. The current programmed I/O interface to the core and scanner was transferred from the PDP8 to the PDP8/e on June 5.

The interface to the 360/75 via the 2701 PDA was designed, wired and checkout started on June 6.

The maintenance processor/Exchange Net interface is a flexible, high-speed interface to be used in debugging the various units of Illiac III. The interface is about 80% designed. The maintenance processor/TP interface has been designed and is being implemented.

The real-time clock, a crystal-controlled clock with program selectable time base, was designed, wired and checkout started. This clock is described in Department of Computer Science File No. 863.

The bus drivers and device selectors, to facilitate interfacing programmed I/O devices, were installed and checked out. These devices are described in Department of Computer Science File No. 862.

4.1.3 Scan-Display Devices

The large format television system was assembled and initially adjusted with good results. The unit consists of a modified graphic arts camera bed on which is mounted a television camera, 2 KW of quarts front lighting, and variable cold cathode back lighting. Both the camera and copy board are movable so that magnification can be changed. Maximum back lighted copy size is about 14 x 17 inches (x-ray film). Front lighted copy can be slightly larger. In terms of size as reproduced at the monitor screen, the magnification range available is of the order 2X to 0.8X (continuous). Lens filters will be used for color investigations.

The remaining television camera was mounted on the automatic microscope unit to be used for remote program controlled operation. All stepping motors for the stage were installed and test run under manual control. A repeatability check showed that the X and Y axis motors relocate the image ± .0002 inches reliably. A 200 step per turn motor will replace the current 24 step per turn unit for Z axis (focus) control.

PDP/8 machine language and FORTRAN programs are currently being written to calculate the Scanner/Monitor performance parameters.

4.2 PARALLEL PROCESSING OF PICTURES

4.2.1 Pattern Articulation Unit

The Pattern Articulation Unit (PAU) is a unique parallel processor used for the analysis of digitized images in planar form (figure 2). It is our intention here to devise, implement and evaluate parallel processing strategies appropriate to the Pattern Articulation Unit of the Illiac III Computer.

The PAU Display System which is capable of displaying the states of the 1024 stalactites in the Iterative Array, was completely debugged and is now operational.

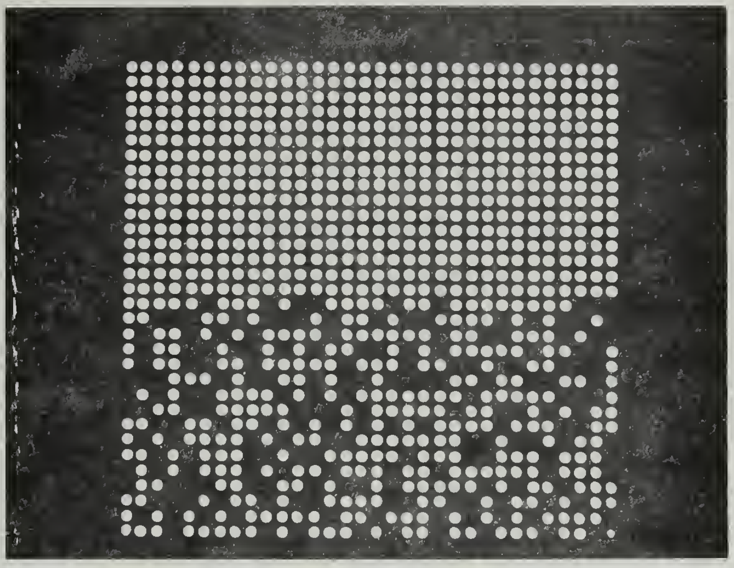
One-half of the Iterative Array (512 stalactites) is now operational. Wiring of the stalactites of the second half have also been completed but not checked out (figure 3).

Two-thirds of the PAU control is now completed. In particular, the GATEIA instruction has been implemented which allows us to simulate by microprogramming the operation of any other instruction.

PAX II Language Support: The first issue of the "PAX Users Group Newsletter" has been sent to all installations known to have a version of the PAX II Subroutine Package. Requests for copies of this newsletter should be sent to PAX Newsletter Editor, Illiac III Project, Department of Computer Science, University of Illinois, Urbana, Illinois, 61801.

4.2.2 Signal Detection Theory

The central recognition task of image processing is the segmentation of pictures by the detection of boundaries and textural differences. Plane parallel picture processing rests or falls on the success of such methods. Accordingly, we place great stress upon the fact that the design of optimal image filters should rest upon a sound methodological basis (e.g. signal detection theory) and move away from entirely ad hoc procedures. Parallel processing algorithms for pictures should admit a technical basis comparable to the classical design principles for linear filter networks.



P3 DISPLAY

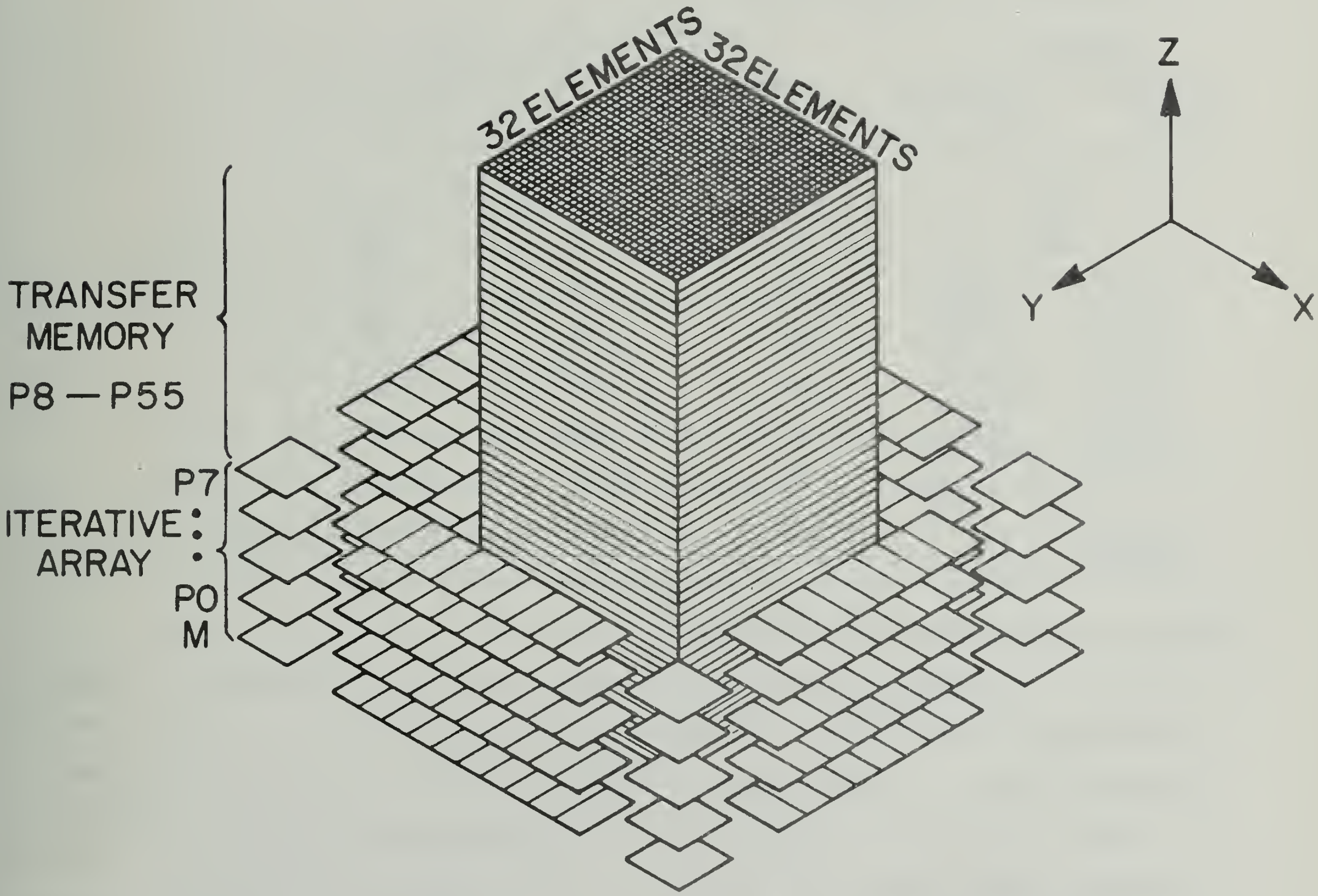


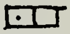
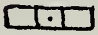
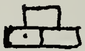
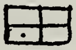
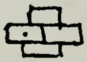
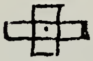
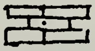
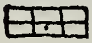
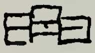
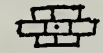
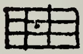
Fig. 2 PATTERN ARTICULATION UNIT

4.2.2.1 Extensions of Signal Detection Theory

Regions of different textures can be differentiated from each other if it is possible to extract the patterns which characterize these textures. Ideally each texture consists of only one single pattern, known as the 'texture element' (unit cell). But extracting the 'texture element' (whose shape and size is generally unknown) in a natural texture is a formidable task. Instead we can define a complete set of patterns and search for the occurrence of these patterns in the given textures.

4.2.2.2 Pattern Elements

We define a template centered around a point which extracts the gray levels of its neighboring points. All possible patterns define the universe of texture elements. A table showing the template and the number of elements in the universe is shown below.

Template		Number of gray levels in the picture			
Regular Mode	Hexagonal Mode	NGL = 2	NGL = 3	NGL = 4	NGL = 16
		2	3	4	16
		4	9	16	256
		8	27	64	4,096
		16	81	256	65,536
		32	243	1,024	1,048,576
		64	729	4,096	17M
		128	2,187	16,384	268M
		512	19,683	262,144	68,600M

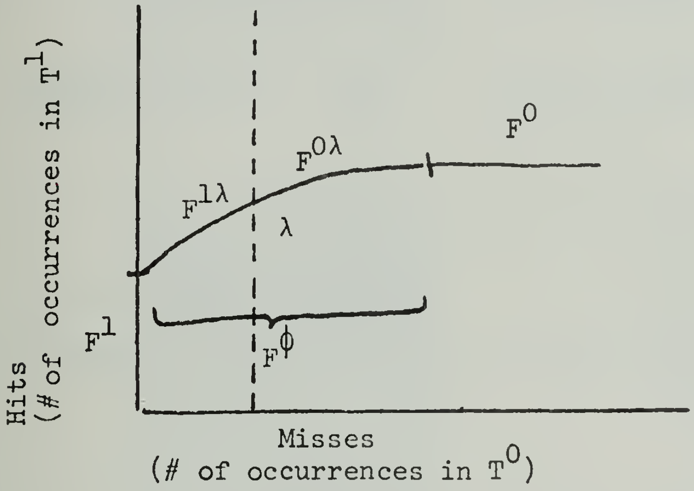
4.2.2.3 Development of the ROC Curve

Considering the n-tuple of gray levels from each sampling template as an 'event',a ROC (Receiver Operating Characteristic) can be built up to discriminate between a pair of textures, say T^1 and T^0. Let E^1 and E^0 be the set of events obtained from scanning T^1 and T^0 with a given template respectively. Using this information, the universe of events (E_U) is divided into 4 disjoint classes, F^1, F^ϕ, F^0 and F^*. F^1 and F^0 are the events which occurred exclusively in T^1 and T^0 and F^ϕ consists of events which occurred in both. F* contains events which occurred in ncne of these scenes. Defining the 'likelihood ratio' of an event as

$$\text{L.R.} = \gamma \frac{\#\text{ of its occurrences in } T^1}{\#\text{ of its occurrences in } T^0}$$, where γ is a

normalization factor to compensate for intrinsic probabilities, the L.R. of each event in F^ϕ is calculated and these are ordered according to L.R. with the event with the largest L.R. as the first element.

Three sets of events F^1, F^ϕ (ordered) and F^0 can be represented graphically as shown. The graph is called the R.O.C. Curve [2]. Selecting a 'suitable' break point λ, F^ϕ is divided into two classes $F^{1\lambda}$ and $F^{0\lambda}$.



Now consider the set union $F^1 \cup F^{1\lambda}$. This is a set of events whose occurrence is more probable in T^1 than in T^0. Similarly with $F^{0\lambda} \cup F^0$.

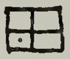
After obtaining these two sets, we return to T^1 and T^0 and mark differentially the events which are members of the sets $F^1 \cup F^{1\lambda}$ and $F^{0\lambda} \cup F^0$ respectively. We obtain in this manner two 'colored' pictures which are more easily discriminated than the original pictures. If we repeat this process, we commonly obtain yet better results and often find that T^1 will be colored completely with one color and T^0, another.

4.2.2.4 Detecting Textural Boundaries

Given the proto-samples of a texture present in a combined scene, borders between these textures can be detected using the ROC technique. Here we use the combined picture as T^1 and the collection of isolated proto-samples as T^0. Here the equivalence class F^1 which contains the events which occurred in T^1 alone, will predominantly contain events which occur on the border. If we go back and detect all the events in F^1, we normally are able to extract the border. This experiment has been conducted using a 2 x 2 window on artificially created textures and natural textures, i.e. 'grass' and 'stone' from Broadatz's book of textures . Results are illustrated below.

Example 1: Artificial Texture

A deformation in a regular design is detected.

Template defining the event = 

Gray levels = 4

Total number of events in the universe = 256

Typical Distribution of Events:

Set	# of elements
F^1	3
F^ϕ	5
F^0	0
F^*	248

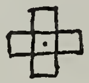
Two iterations of the filtering process are applied to give the boundary identification shown in figure 4.

Example 2: Grass and Stone

Stone boundaries are stripped from a composite picture containing both textures (see figure 5).

T^1 : Combined picture

T^0 : A window from 'grass' and a window from 'stone'.

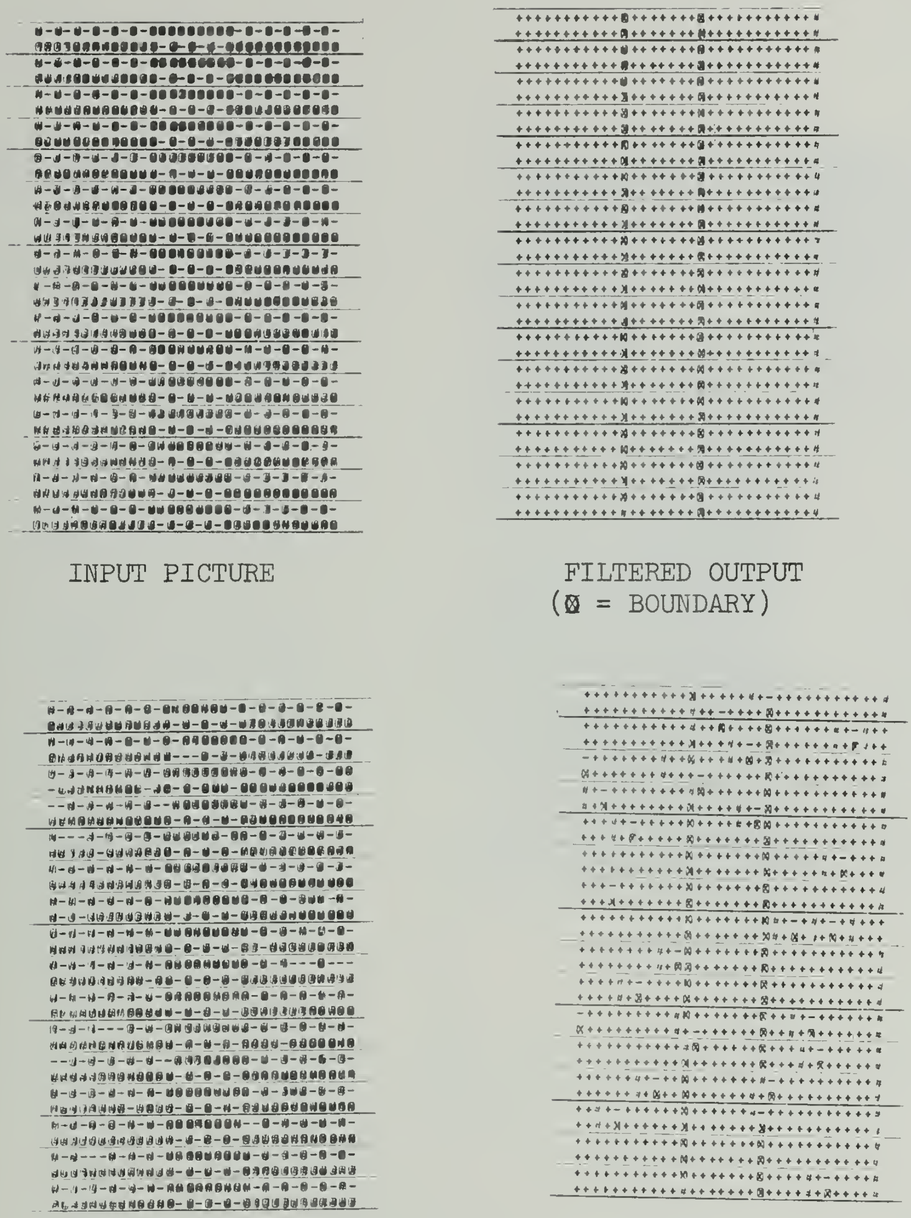
Template defining the event = 

Gray levels = 4

Total number of events in the universe = 1024

Typical Distribution of Events:

Set	# of elements
F^1	93
F^ϕ	138
F^0	36
F^*	757



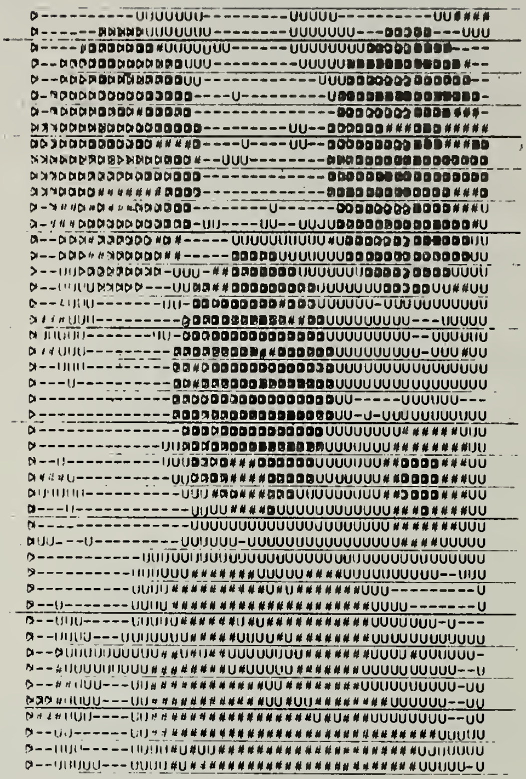
INPUT PICTURE

FILTERED OUTPUT
(⊠ = BOUNDARY)

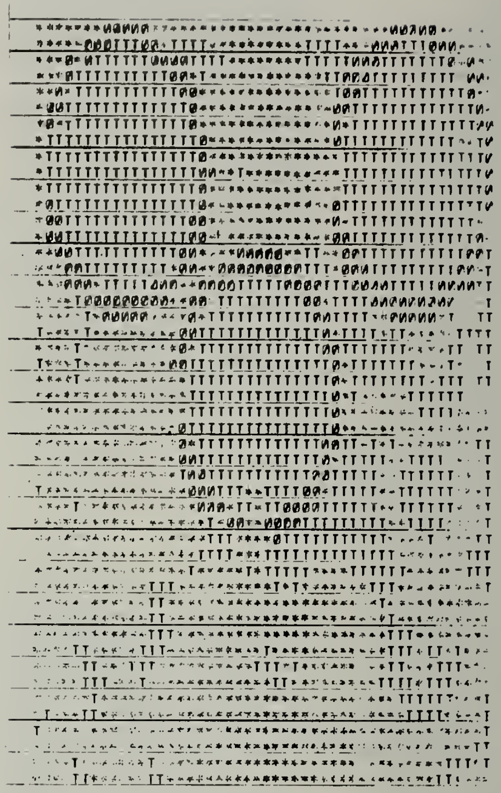
AS ABOVE WITH NOISE

FILTERED OUTPUT

Fig. 4 SIGNAL DETECTION: Example 1



INPUT PICTURE



FILTERED OUTPUT
Ø INDICATES TEXTURE BORDER

Fig. 5 SIGNAL DETECTION: Example 2

4.2.3 Interval Coverings

4.2.3.1 Orientation

In the course of a series of seminars this year (Sight Sensory Systems Seminars, see bibliography) Michalski and McCormick have developed the basis of a new pattern classification procedure. The technique allows one to find an optimum set of k-dimensional "intervals" to package an observed set of k-tuple patterns. The procedures are entirely consistent with the foundations of signal detection theory, and in the limit of adequate statistics gives a class of optimal recognizers corresponding to points on the ROC (receiver-operating characteristic) curve. (See figure 6)

This work, in one sense, generalizes classical switching theory concepts of a map covering. This correspondence is mutually beneficial, for the combinatorial algorithms of finding an optimum covering can then be adequately resolved using algorithms of known effectiveness from switching theory. Attention in our work to date has centered upon the very effective "method of disjoint stars" developed by Professor Michalski.

4.2.3.2 Formal Description

A report describing the theory has been prepared [3].

The paper considers:

(1) a Boolean algebra $< 2^E, \cup, \cap, -, E, \emptyset >$ of event set E_i from a discrete finite vector space E, and

(2) mappings f from the set E into {[0,1], *}, where * represents some unspecified value. A special case of the above is the Boolean algebra and Boolean functions considered in switching theory, where E is a space of binary vectors and f maps E into {0,1,*}, i.e. into the end-points of the interval [0,1] and *.

A meet semi-lattice of multidimensional intervals (interval complexes) in E is introduced and then the concepts of exact, free, unordered and ordered interval covers of f are defined.

The simplest case of a cover - an unordered exact cover of a set $F^{1\lambda}$ against $F^{0\lambda}$ - is defined as a set of interval complexes whose set-theoretic union covers a given subset $F^{1\lambda} \subseteq E$ (defined as $\{ e \mid f(e) > \lambda\}$) and does not cover any element of another given subset $F^{0\lambda} \subseteq E \setminus F^{1\lambda}$.
(See figure 7.)

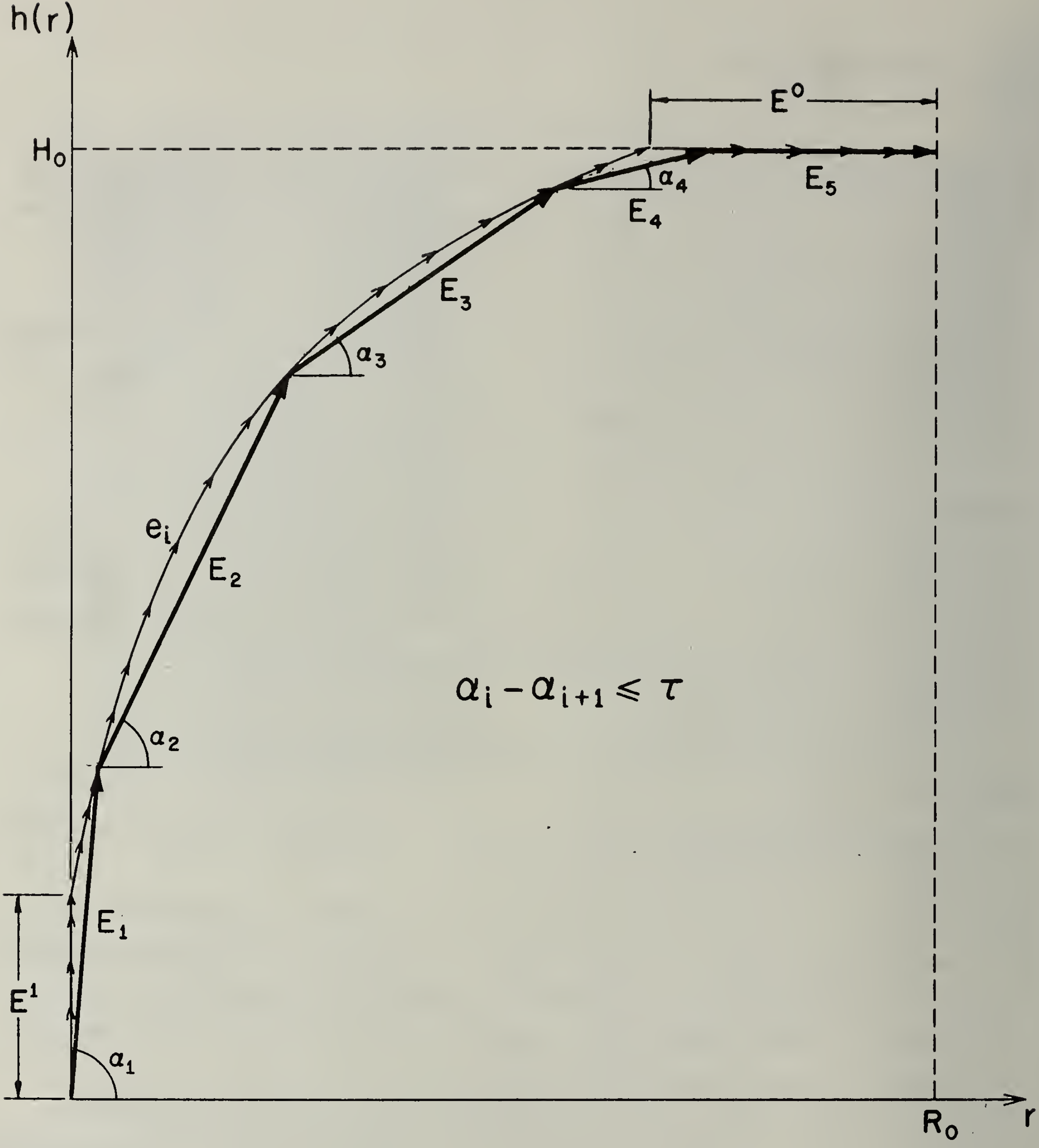
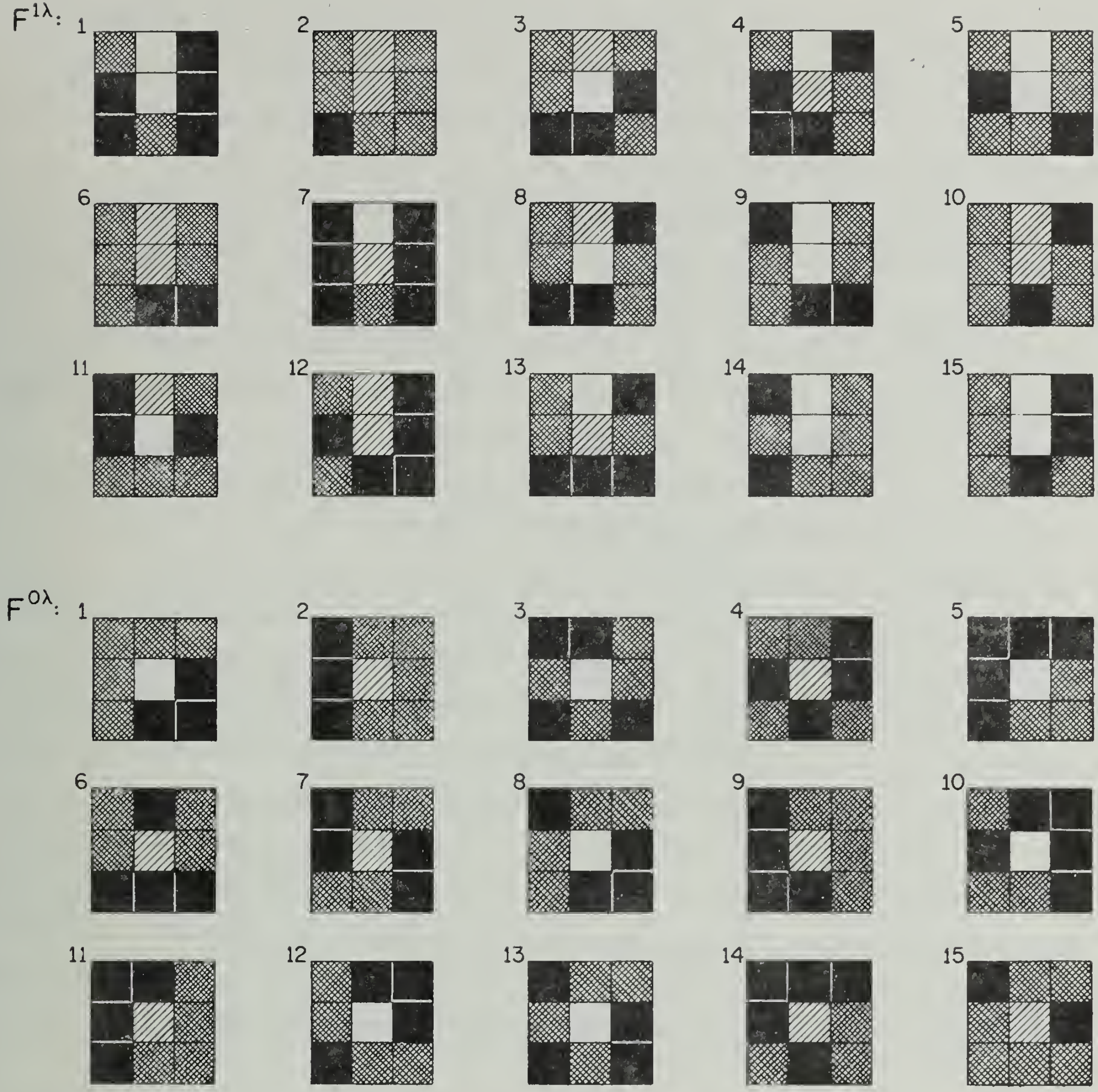


FIG. 6 RECEIVER - OPERATING - CHARACTERISTIC (ROC)

$n=9,\ h=4,\ c(F^{1\lambda})=15,\ c(F^{0\lambda})=15$



$M^q(f/\lambda)$:

$\Delta = 0$

$t = 5.5\,\text{sec}$

*	▨	*
*	*	*
*	*	*

n - number of variables (in our case Number of elements of the square)

h - number of grey levels

* - denotes that the corresponding to square-element variable is not relevent.

FIG. 7 DISCRIMINATION BETWEEN TWO SETS OF PATTERNS

(U-LIKE PATTERNS $F^{1\lambda}$ ARE DISTINGUISHED FROM O-LIKE PATTERNS $F^{0\lambda}$ BY INTERVAL COVERING WHICH EMPHASIZES THAT ONLY ONE CELL IS SIGNIFICANT - SEE $M^q(f/\lambda)$. THIS COVERING WAS FOUND BY OPERATIONAL INTERVAL COVERING PROGRAM

The concept of ordered covers was developed to accomodate a preferential order in covering the set of 'mixed' events, defined as $\{ e \mid 0 < f(e) < 1 \}$ (a case not considered in classical switching theory). (See figures 8, 9, 10).

The synthesis algorithm of covers is based on the 'method of disjoint stars', which has proved to be very useful for synthesis of complex switching systems. Quasi-minimal covers, produced by this method, are either minimal or approximately minimal. However, when we cannot state that the obtained solution is minimal, an estimate of its maximal possible distance to the minimum is provided.

Applications of the interval coverings concepts to pattern recognition and picture filtering are delineated.

4.2.3.3 Remarks on Applications

A cover of a mapping f consisting of multidimensional intervals can be interpreted as a set of 'filters' for recognizing events from a signal class (represented by $F^{1\lambda}$) for pattern recógnition and picture processing purposes, e.g. for discriminating regions of different textures, striping of background, local feature extraction, border detection, etc.

In case of an ordered cover the individual filters correspond to the consecutive points on the optimum receiver-operating-characteristic (ROC), defined as in statistical decision theory. The above ROC curve can be obtained by optimal ordering events from the two classes to be distinguished with regard to the likelihood ratio of their frequency occurence.

X_1	X_2	X_3 = 0				X_3 = 1				X_3 = 2			
	X_4	0		1		0		1		0		1	
	X_5	0	1	0	1	0	1	0	1	0	1	0	1
0	0						0						
	1			1		1		1			1	1	1
	2		0		0	1		1			1	1	1
1	0							1		0			1
	1			1		2		1	3		1		1
	2	0	3		3		2	2	3				
2	0					0			2		3	2	
	1		2					1	0			2	2
	2				3		3		2			2	2
3	0	0					3			3	3	2	
	1				2		3						
	2	0		0		0	3						

FIG. 8 IMAGE T(f)

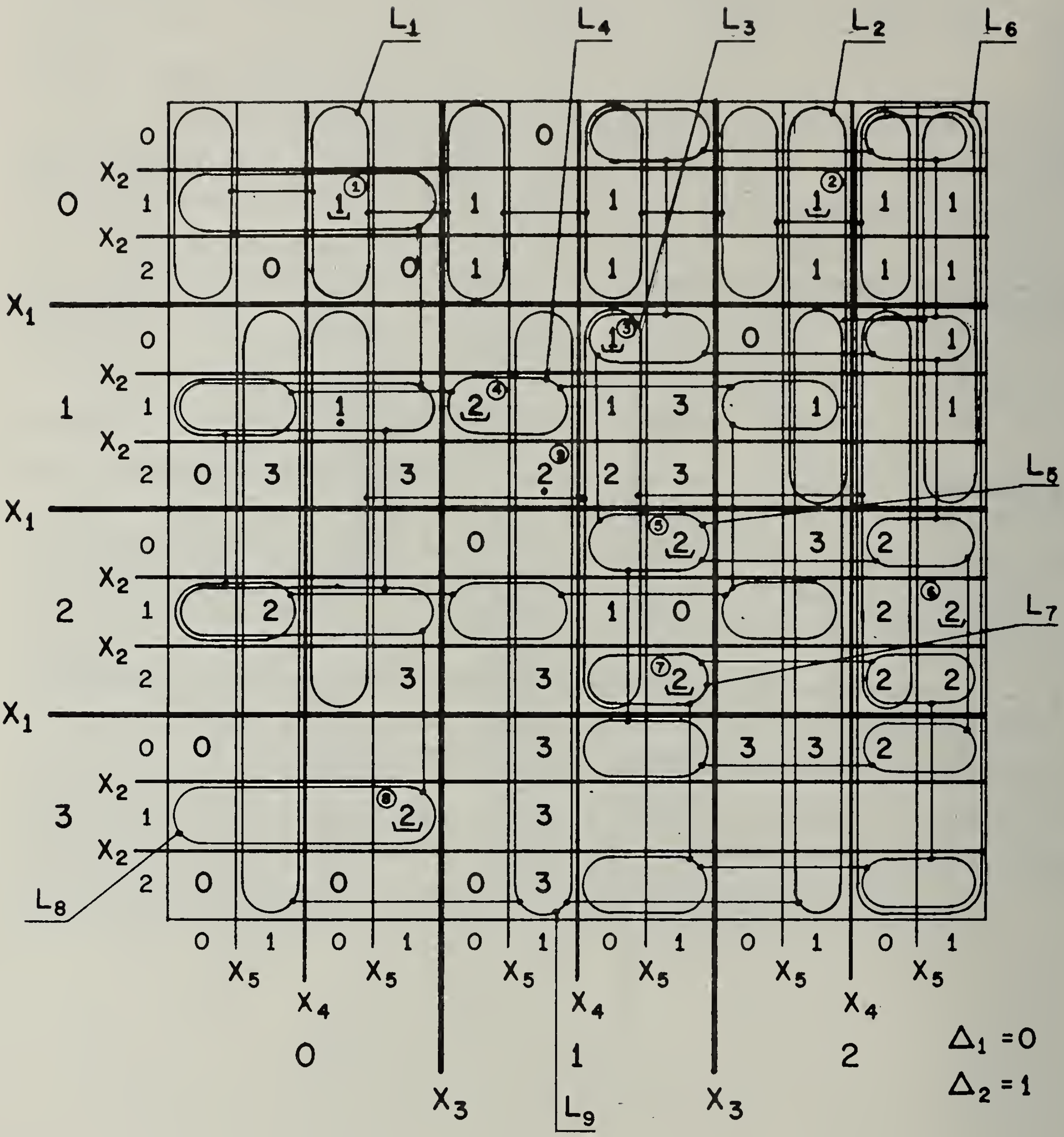


Fig. 9 ORDERED INTERVAL COVERING OF IMAGE T(f)
(Events numbered 1 are covered first, then 2, etc. Compare with Fig. 5)

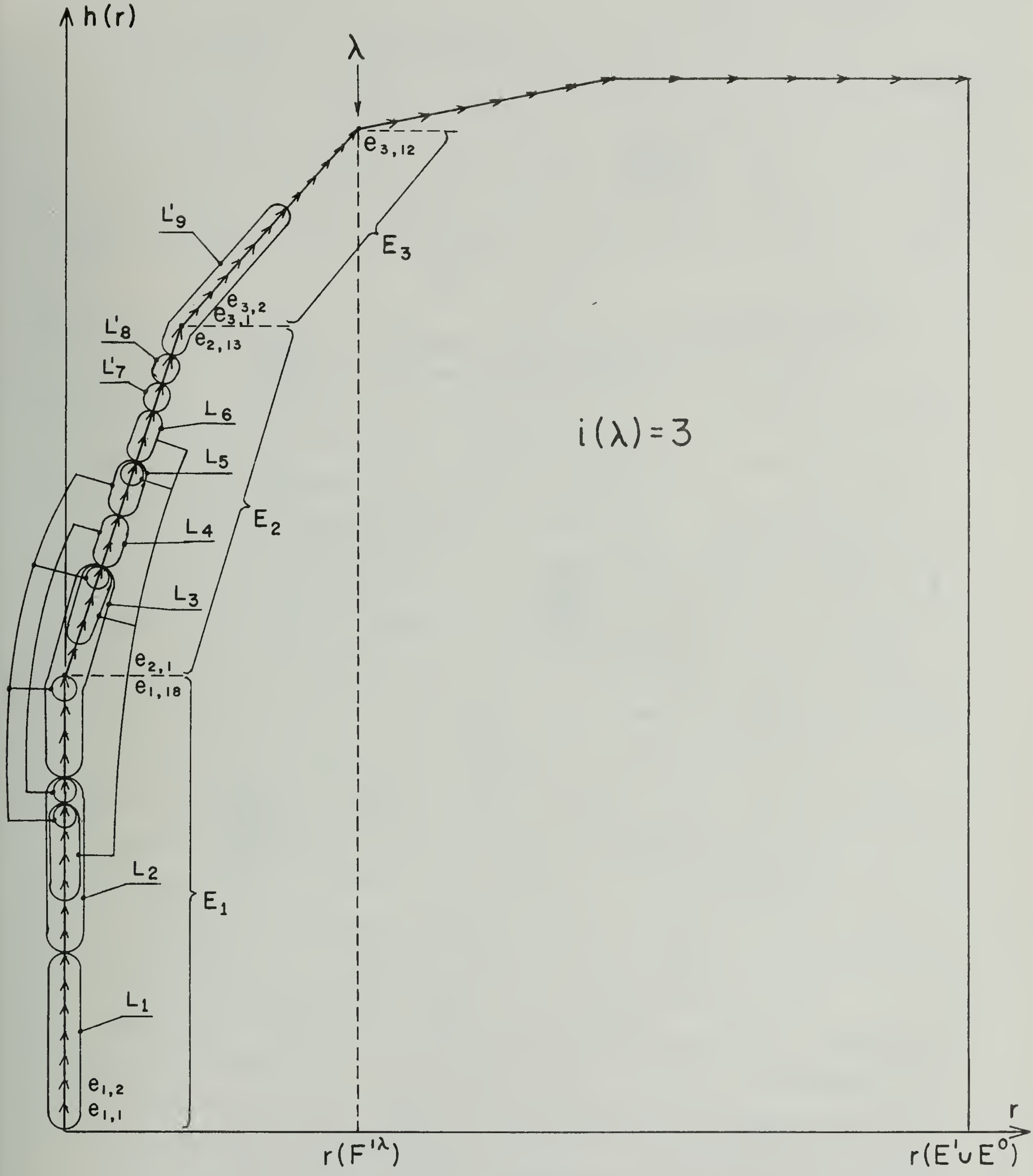


Fig. 10 COVERING INTERVALS REFERRED TO THE EVENT SETS OF THE ROC CURVE (SEE FIG. 4)

4.3 STRUCTURAL INFERENCE

Our long range goal can be viewed as this: to design a sight sensory system which can infer the syntactic and semantic structure of its visual environment from selected instances of objects and scenes from the environment. Implicit here are parallel processes for scene segmentation based on the signal detection theory and appropriate generalizations of covering theory. Through dialog with the system, a user may participate in the acquisition of the structural description of the visual environment.

4.3.1 Scene Segmentation

The picture processing model attempts to develop the structure and semantics necessary to treat a general two-dimensional scene. To achieve this goal, we introduce the first characteristic of the model. The picture is initially mapped into a cellular mosaic: a segmentation of the scene into cells (e.g. regions bounded by simple curves) which then form primitives for further analysis. (See figure 11 A & B) The choice of primitives strongly influences the development and the capability of the model. Cells seem to be far superior to primitives consisting primarily of line-elements, which were used in earlier approaches. Line-elements were often chosen to attempt to obtain a model based on one-dimensional and context-free grammars, which are inadequate in many cases.

Scenes are represented internally in the system as a hierarchical structure of cellular mosaics. The cells, or regions, of the mosaics, considered as nodes with assigned attributes and names, are linked (according to selected binary relations) to form a labelled digraph representation of the scene. The decision rules for when to select a given binary relation or to classify two regions as having analogous attribute values are inferred by the system from instances of the visual scene in the instructional period.

The graph representation is chosen for global level picture processing. The recognition process can then be viewed as replacement rules operating on graphs. This process, which is analogous to parsing, is called "composite formation." A replacement rule forms a composite element from subelements of the graph. The new graph formed by new composite elements and new relations involving these elements is then

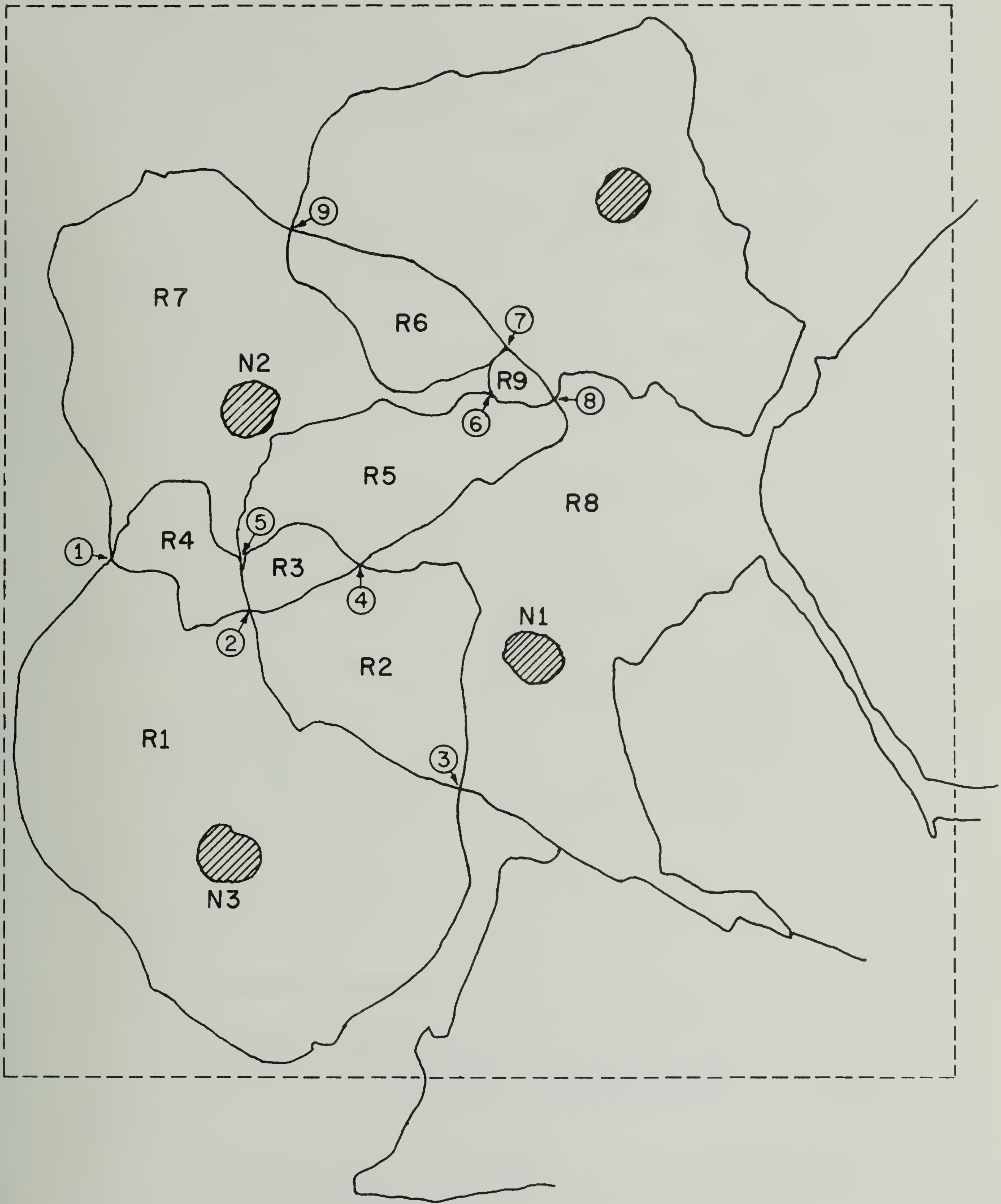


Fig. 11A PARTIAL MOSAIC REPRESENTATION.

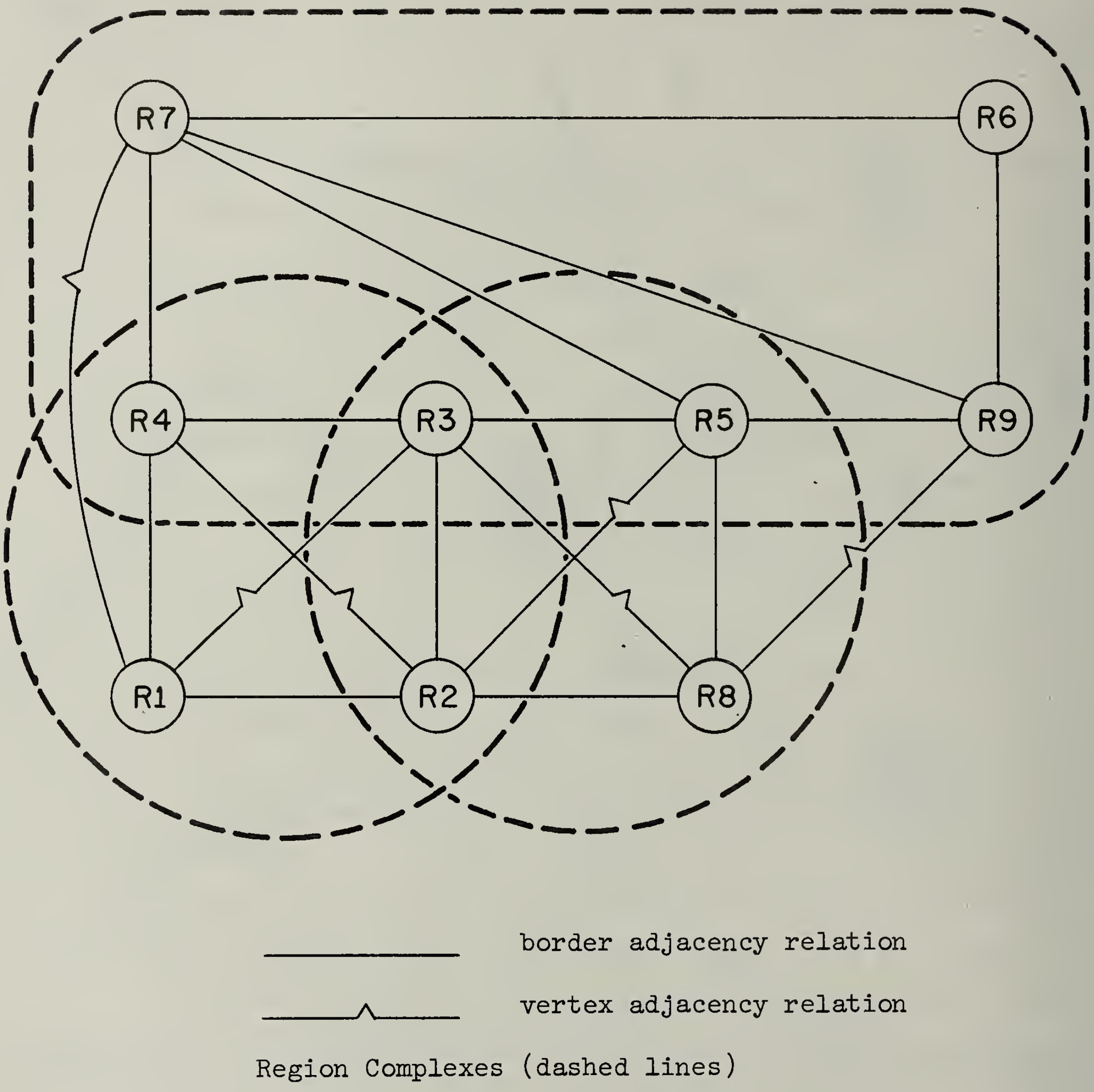

R7
R6
R4
R3
R5
R9
R1
R2
R8
border adjacency relation
vertex adjacency relation
Region Complexes (dashed lines)

Fig. 11B COVER FOR MOSAIC

the candidate for further composite formation. For successful recognition, this process should tend toward a "recognizable graph," (see figure 12) i.e. a graph which has somehow been specified as an acceptable goal.

The key property of the system is its ability to rapidly infer which factors of the binary relations and which multidimensional intervals of attribute values best articulate the scene for a given mode of system behavior and visual environment.

Toward the end of the current contract year, a thesis by John C. Schwebel will summarize the larger part of our work to date in this area. In addition, Michalski and McCormick are preparing a paper for the Paris meeting of IFAC Conference.

4.3.2 Structure Operation Language Support

We have specified a Structure Operation Language , SOL [4] in order to be able to simply and conveniently express operations on multigraph structures which allow an arbitrary number of variables to be associated with any network element.

To justify the definition of yet another "programming language" and put its development into perspective, we cite the following facts: We are currently able to program picture processing algorithms which operate on the PAX plane representation of pictures as binary valued elements with neighborhood connectivity relationships. The next and most natural abstraction from a PAX plane representation is a graph-strucutre, (i.e. a hierarchical system of graphs). Many scene segmentation algorithms can be expressed most simply by operations on a graph structure. We have, and are continuing to develop, theory for the next higher level of picture processing operations represented as structure transformations.

Thus, the structure operation language is quite helpful for precisely specifying and for experimenting with heuristic scene segmentation strategies.

The language has a small number of root operations which are sufficient for expressing network transformations and has been specified with the goal of embedding it in PL/1. Figure 13 lists and illustrates the root operations of the language.

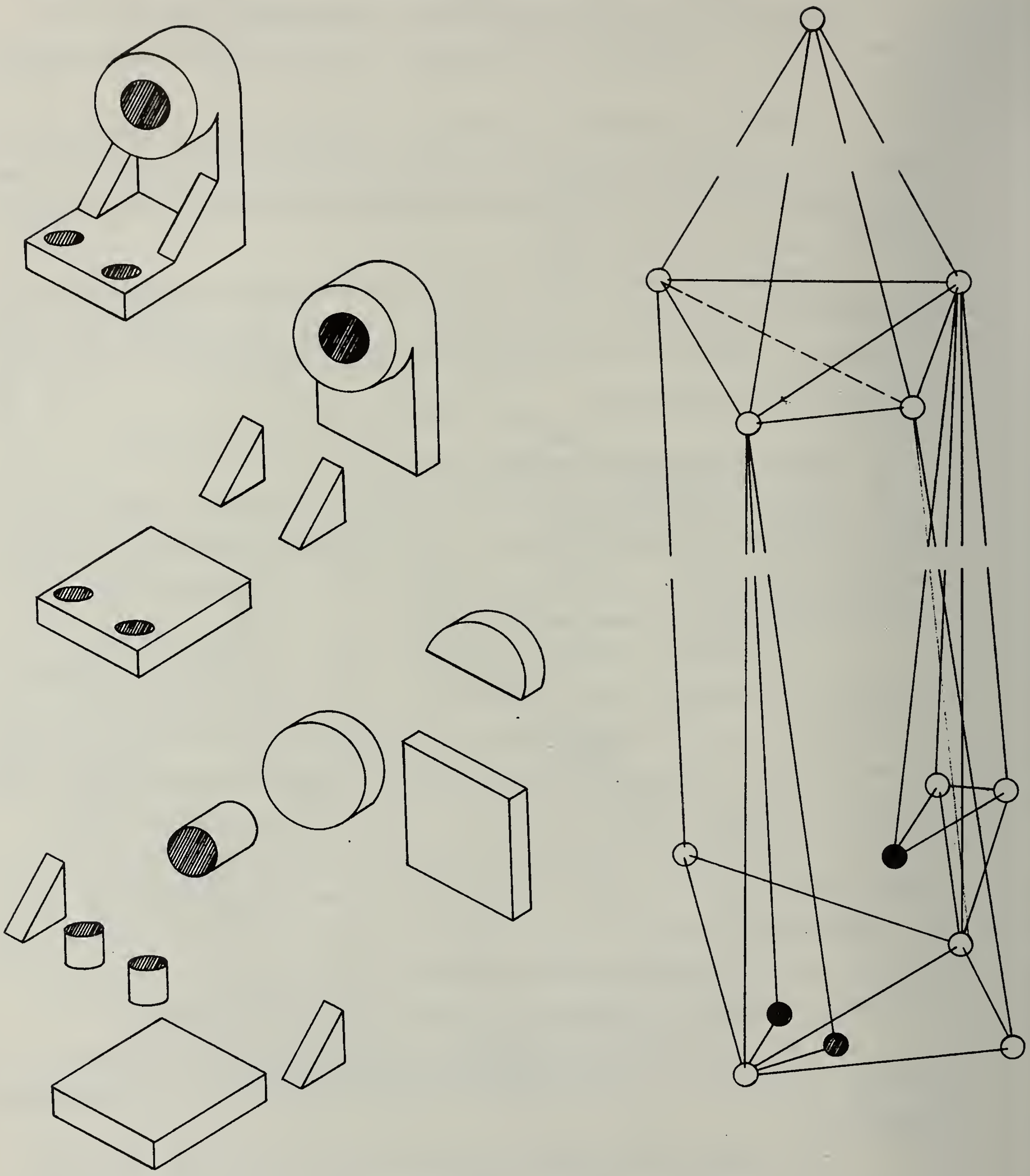
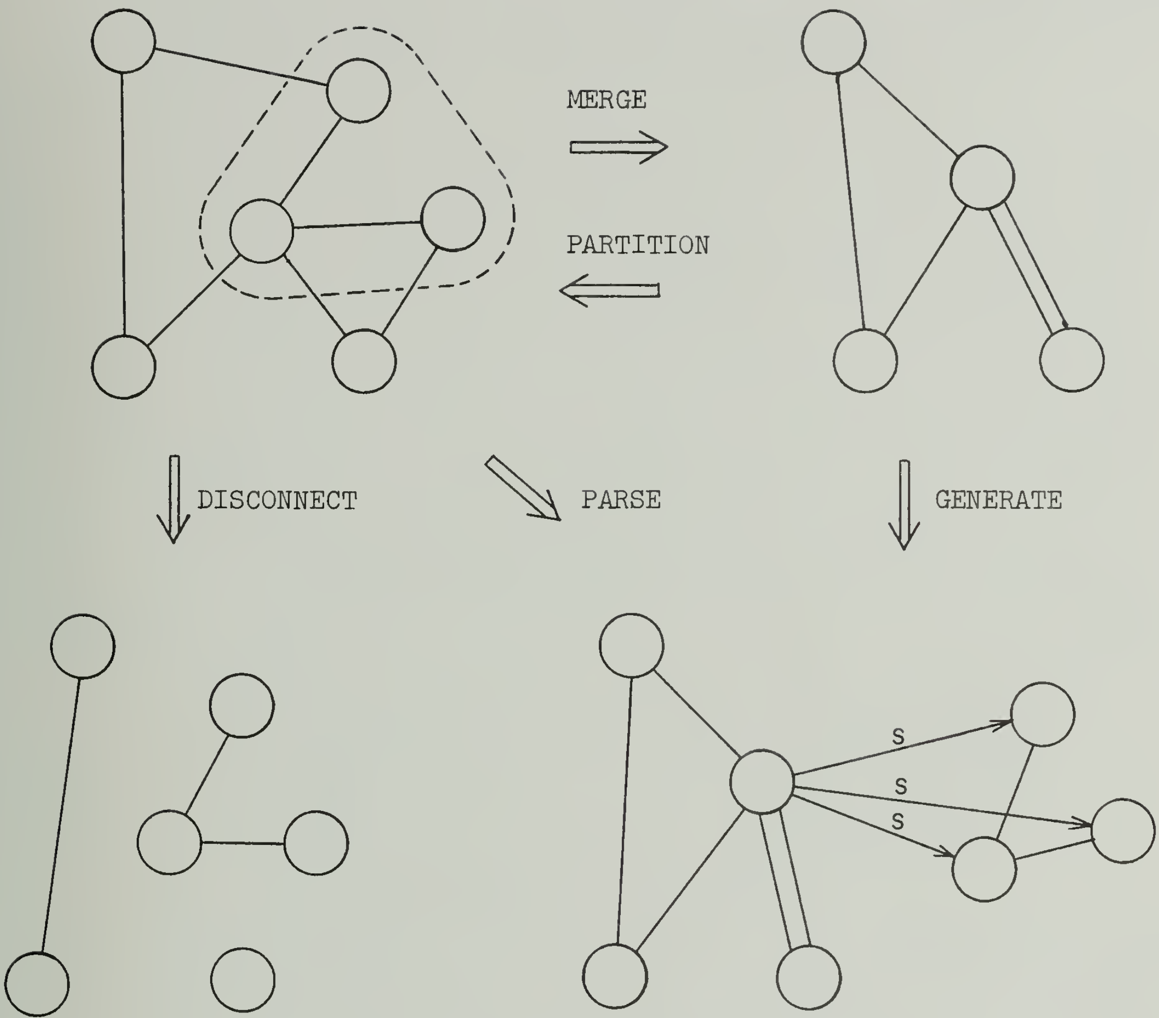


Fig. 12 COMPOSITE FORMATION



a) subgraph operations

LINK	subgraph	SELECT	element
MOVE	pointer	DECLARE	element
ADD	element	ASSOCIATE	variable
DELETE	element	FREE	variable

b) other operation types and operands

Fig. 13 SOL OPERATIONS

The language is still in the early stages of implementation.

A syntactic analyzer for the language is under development using the Translator Writing System of Alan J. Beals, Department of Computer Science, University of Illinois.

4.4 APPLICATIONS

The operational tools, hardware and software to begin a significant attack on several applications were begun. Automated cytology was emphasized this past year, as cells with their relatively simple morphology provide a ready short range target for the parallel image processing strategies of the Illiac III computer system.

4.4.1 SEM Micrographs

The Scanning Electron Microscope has been extensively employed in its applications to material science as well as subcellular biology. A JSM-3 SEM, Japan Electron Optics Laboratory Company was installed in the Material Research Laboratory, University of Illinois, Urbana, last summer. Mr. John B. Woodhouse of MRL is in charge of the facility. With his help, a program to explore the benefits of image processing automation for SEM imagery was initiated. The system design of an automated on-line scanning electron microscope has been in process.

One of our main goals is to study cells and subcellular populations as generated by the cytospectrometer. We have undertaken two preliminary tasks:

(1) Slide preparation: With the services of Mrs. Beth Lepinski and Mrs. Grace Conway of Mercy Hospital, Urbana, Illinois, we have prepared a variety of samples such as red cells, leucocytes, urine crystals, bacteria, chromosomes, etc.

(2) Evaporated coating techniques: The problem arises from preparation of a slide to be available to be observed both under conventional optical microscope and SEM. With the help and suggestions of Professor Francis Young of Civil Engineering Department, University of Illinois, Urbana, a thin coating of 200 Å of carbon and Au/Pt alloy has been tried as a compromise to solve this problem.

In addition a special specimen carrier for integrated circuits has been constructed, so that the circuit can be operated within the vacuum chamber of the SEM. Checkout of LSI circuitry of the future will likely entail the use of an electron beam probe (i.e. the scanning electron microscope) quite as today the technician uses a manually positioned contact probe. Our studies here are still preliminary.

4.4.2 Cervical Smears

A central problem in the construction of machines to automatically inspect and classify samples of cervical cells (e.g. Pap smears) has been the tendency of the cells to clump, fold and overlap in such a way as to confound the analysis algorithms proposed. This problem occurs whenever morphologically-based measurement is attempted on cells of this type, even after they have been suspended in a fluid medium and mechanically rearranged with flow or droplet techniques.

While non-morphological measurements such as total ultraviolet absorption or total light scatter* may prove to be sufficient for the cervical cell screening problem, it is felt that the ability to reliably locate and identify various cell types and substructures in the face of folding and clumping provides a firmer base for development of new techniques for automated cytology.

It was decided here to proceed by attempting to develop techniques to segment microscope slides of cervical cells prepared according to the standard method of Papanicolaou. Work on this problem has indicated the feasibility of using the Illiac III system organization to do digital processing of Pap smears without disrupting the form of the sample currently used by pathologists, an approach which reduces many difficulties in correlation of results and establishing follow-up of positives which arise when the cells are transported in other ways. In addition it is anticipated that feature extraction and pattern articulation procedures developed for this application will transfer easily to other cytology applications.

The picture processing strategy adopted here has been to approach the image in a top-down fashion because of the fairly limited number of different forms to be located, and because of the simple structure of the forms, which consist basically of blobs (dark, compact, symmetric regions) and conglomeration of blobs of various sizes and textures. For example,

* Kamentsky, L. A., and Melamed, Instrumentation for Automated Examination of Cellular Specimens, Proceedings of IEEE, Vol. 57, No. 11 pp 2007-2016.

large blobs with a coarsely grainy texture have a high probability of being malignant cell nuclei. Some preliminary results are shown in figures 14 through 17E. Figure 14 shows a photograph of a field of a Pap smear [60x]. Figure 15 shows the 128 x 128 element digitization of the field. (The 46 mm film scanner is being used temporarily while the microscope scanner is under checkout.) A filter was applied to this picture which detects blobs of a size just greater than a white cell (small dark spots in figure 16) while rejecting blobs composed of white-cell-sized subblobs. The output of this filter, shown in figure 16 is thresholded and locations of high filter output (figure 17A) are rescanned at higher resolution for texture analysis. (See figures 17B, C,D,E.)

The above investigations, the subject of a Master's thesis, are only preliminary and will be continued during the forthcoming contract period.

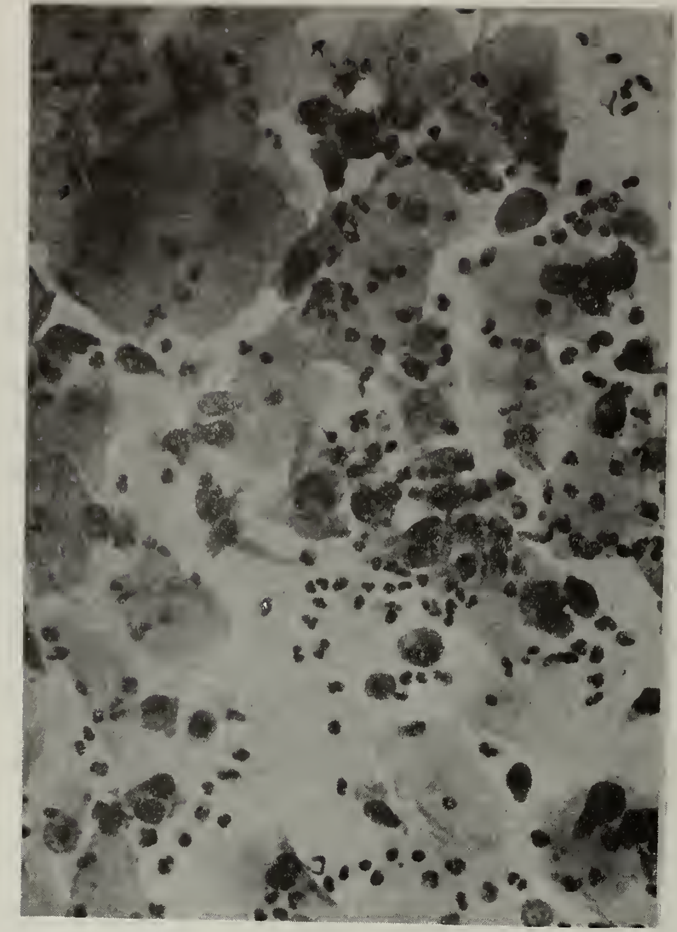


Fig. 14 PAP SMEAR FIELD

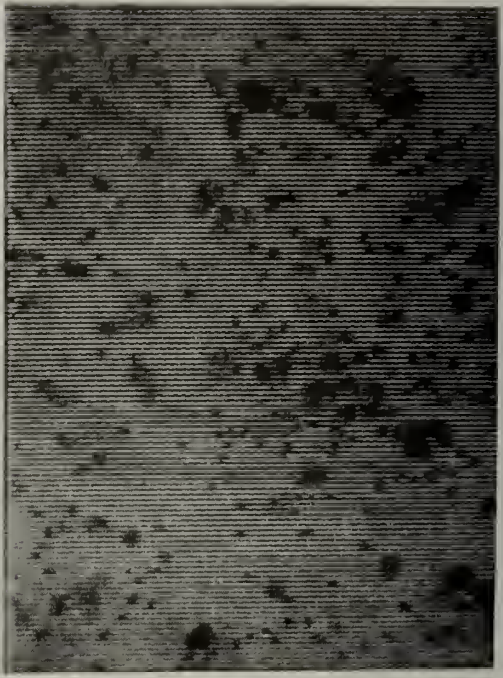


Fig. 15 128 x 128 DIGITIZATION OF FIG. 14

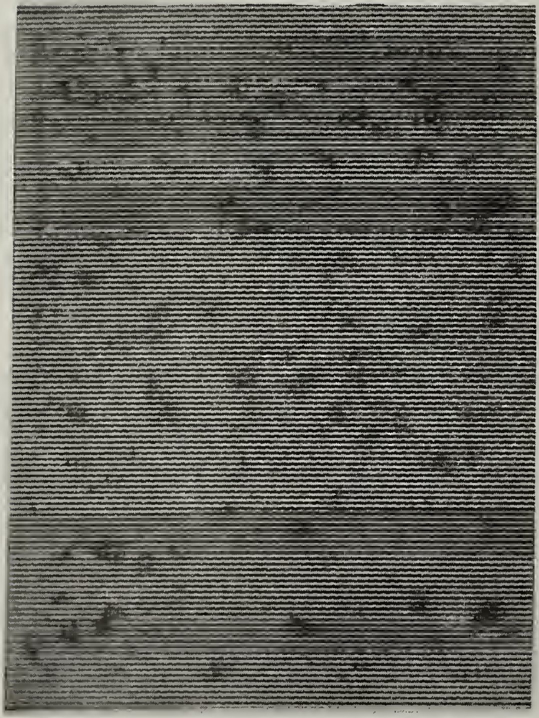


Fig. 16 BLOB FILTER OUTPUT

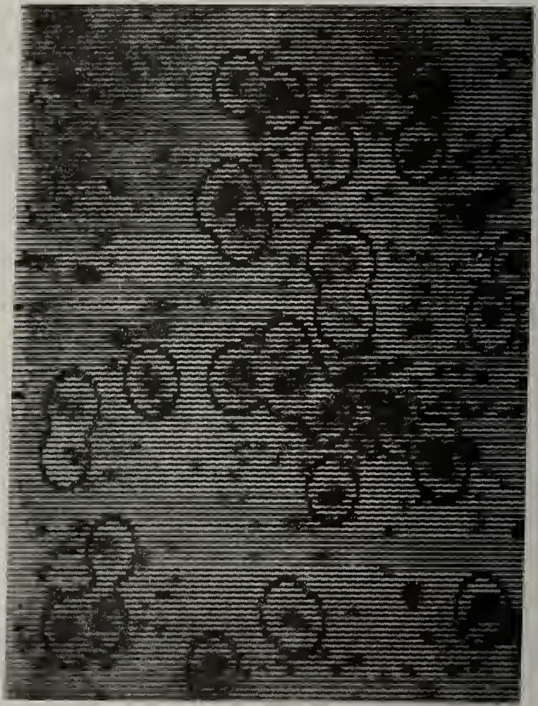


Fig. 17A BLOB HITS CIRCLED

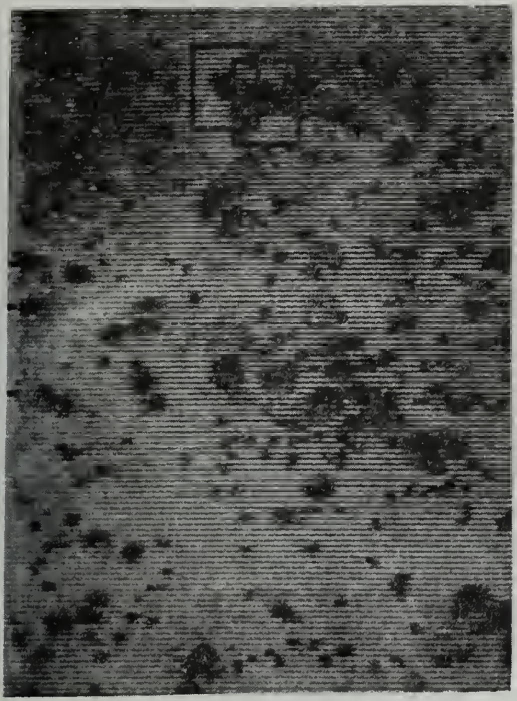


Fig. 17B SECOND RESCAN AREA

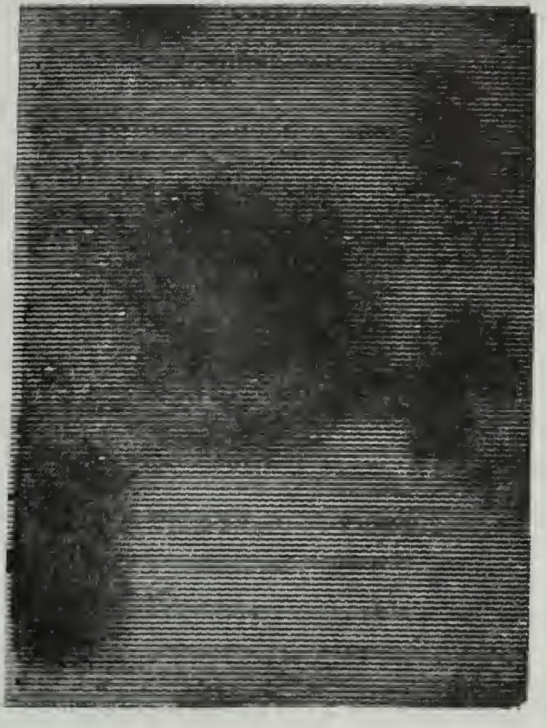


Fig. 17C SECOND RESCAN

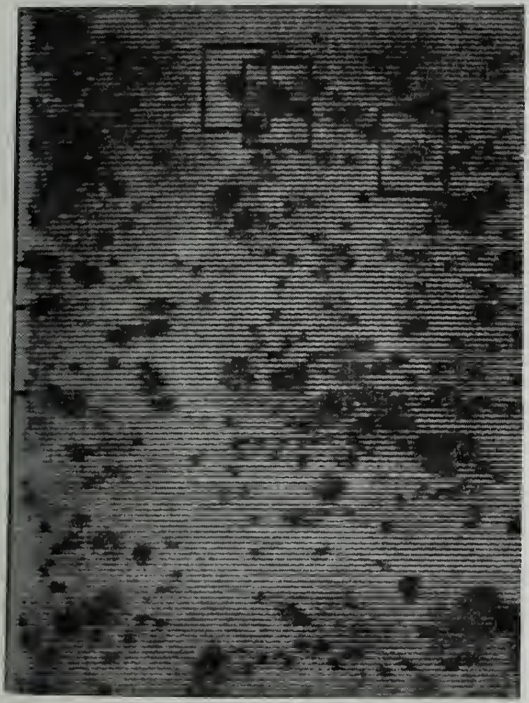


Fig. 17D THIRD RESCAN AREA

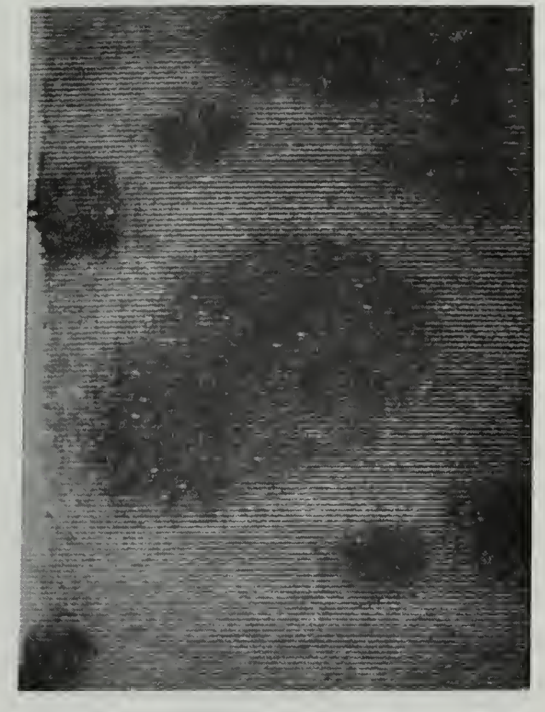


Fig. 17E THIRD RESCAN

4.4.3 Brain Mapping

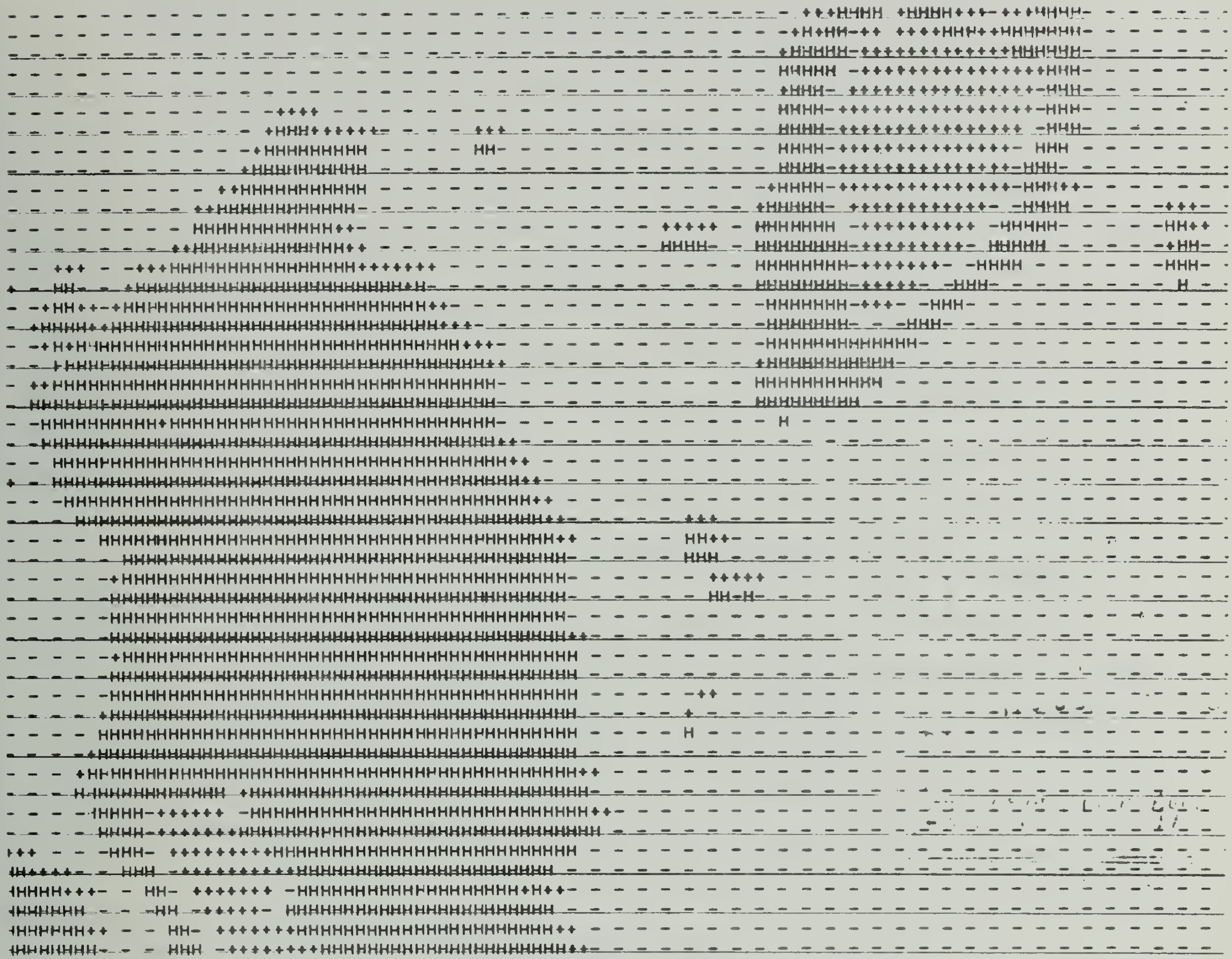
The purpose of this investigation is to develop structural inference techniques and to apply and test them in the environment of neuroanatomy of the brain.

It is intended that these techniques be applied first to a small sample populations of objects in the brain. In connection with current research in quantitative neuroanatomy*, methods are being developed for the automatic recognition of cellular and subcellular constituents rendered visible in cresylecht violet stain and Weil stain under a light microscope. An immediate application is to determine quantitative properties such as the number of neurons in specific domains of the brain and the geometrical dimensions of their nuclei and nucleoli. (See figure 18.) A further application is the delineation of the boundaries of larger domains in the brain, e.g. nuclei, fiber tracts, such that finally a brain map is obtained.

Our approach consists of a preprocessing and an inference phase. In the preprocessing phase, the picture is scanned (i.e. transformed into a grid of points differing in gray scale) and partitioned into closed regions. Two methods to obtain regions are presently being investigated. A region is determined as a connected set of grid points such that (1) they have almost uniform distribution of gray scale values or (2) they form a more complicated characteristic texture. After partitioning a picture into regions, we associate with each region applicable attributes and their values as well as binary relations (containment, adjacency, etc.) valid between pairs of regions.

A simple context-free grammar has been developed whose nonterminal values are class names used in neuroanatomy to designate objects such as cell-types, constituents of cells, etc. The variables are associated with attributes which in turn are connected by semantic rules. Using clustering and covering techniques that are already available or being developed, a variable of grammar is assigned to each region. The structure of the picture is then described by a derivation tree whose nodes are associated with each identified region.

Fry, William J., Quantitative Delineation of the Efferent Anatomy of the Medial Mammillary Nucleus of the Cat, J. Comp. Neur. V. 139, No. 3, July 1970, pp 321-336.



'Fig. 18 FILTERED IMAGES OF BRAIN NUCLEI
(Codes to extract parameters
such as maximum diameter,
perimeter, etc. have been written)

4. 4.4 Cytospectrometer

A new and fundamental technology for the identification and sorting of cells and subcellular components is suggested. A principal ingredient of this technology is the transport of charged particles (cells, subcellular components or macromolecules) by an iterative system of electrostatic quadrupole lenses. These techniques, though extensively used in accelerator installations for the transport of nuclear and high energy particle beams, have not had comparable application to the guidance, identification and separation of macromolecular and cellular particle beams. It is the intention of this research to explore this potential and to devise a first working instrument (called here a "cytospectrometer") which can bridge the gap between these digitally-oriented beam techniques and more conventional methods in cytology emphasizing light and electron microscopy.

4. 4.4.1 Objective of a Cytospectrometer

To provide a two-dimensional spatial separation of constituent particles (cells, subcellular components or macromolecules) on the basis of a 2-parameter differentiation of the incident stream of isolated particles. Parameters available for selection include mass, charge, shape, differential spectral absorption/scattering, rate of discharge in an ionizing atmosphere, etc. The procedures used ideally should maintain the identity of subpopulations, such as all chromosomes from a given nucleus.

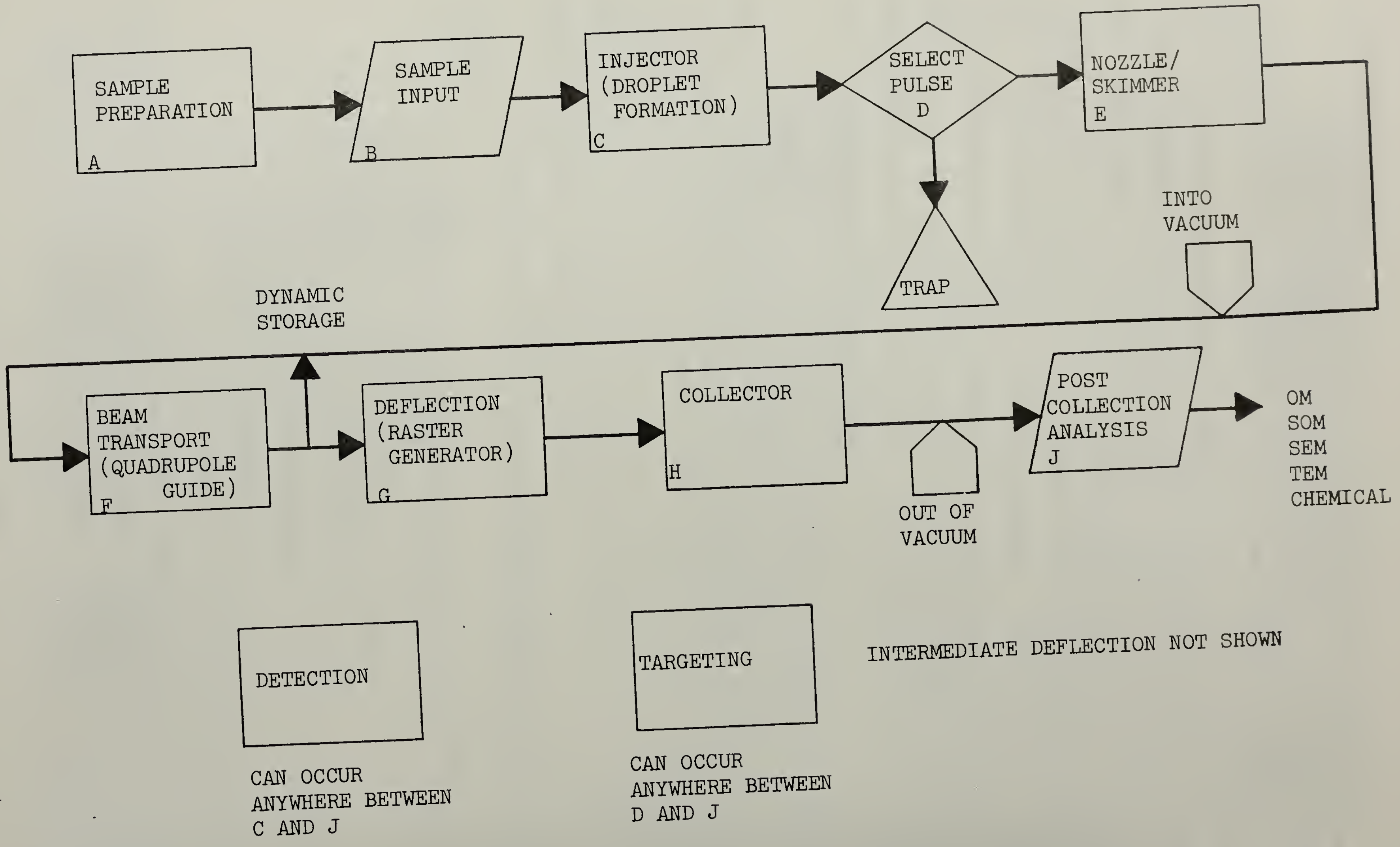
Applications of the cell beam technology that warrant investigation include:

i) Applications to hematology (differential counts, etc.)
ii) Pap smear classification
iii) Separation and classification of bacteria
iv) Isolation of genetic mutants
v) Subcellular particle separation: e.g. cell nuclei differentiation
vi) Karyotyping of chromosomes
vii) Cancer cell discrimination, including possibly stem cell identificat

These potential applications cannot be adequately investigated, obviously, with the level of support requested here, but we should be able to evaluate the feasibility of applying the beam transport technology to these areas.

Constituent subassemblies of a cytospectrometer facility are shown in figure 19. A digitally controlled hypodermic syringe is used tc inject a liquid jet of particles suspended in a solute carrier (figure 20).

It will be critical to the success of the spectrometer system that the features extracted from an individual cell passing through the beam transport system reflect the internal structure of the cell and/or subcellular particle. For example charge or mass gives an obvious discrimination by size (and in particular, specific charge alone almost allows a complete differentiation of white blood cells). These parameters however, do not afford the same sensitivity as histochemical reactions monitored by optical transmission. In particular we view the droplet stream of cells as a potential candidate for ultra-rapid low resolution (certainly not in excess of 32 x 32) image processing. The requirements here, $\sim$ 10 - 100 μ sec. per frame, transcend all known systems, but would appear technically attainable. We anticipate that the design of this on-line system will evolve from our experience in processing cells off-line with the Illiac III Computer System.


SAMPLE PREPARATION
A
SAMPLE INPUT
B
INJECTOR (DROPLET FORMATION)
C
SELECT PULSE
D
TRAP
NOZZLE/ SKIMMER
E
INTO VACUUM
DYNAMIC STORAGE
BEAM TRANSPORT (QUADRUPOLE GUIDE)
F
DEFLECTION (RASTER GENERATOR)
G
COLLECTOR
H
OUT OF VACUUM
POST COLLECTION ANALYSIS
J
OM
SOM
SEM
TEM
CHEMICAL
DETECTION
CAN OCCUR ANYWHERE BETWEEN C AND J
TARGETING
CAN OCCUR ANYWHERE BETWEEN D AND J
INTERMEDIATE DEFLECTION NOT SHOWN

METHODS: CHARGE
MASS
TIME-OF-FLIGHT/POSITION
LIGHT,TRANSMISSION/REFLECTION
FLUORESCENCE/TRACER
ELECTRON SCATTERING
IONIZATION

Fig. 19 PROTOTYPE CYTOSPECTROMETER BLOCK DIAGRAM

Fig. 20 CYTOSPECTROMETER DROPLET STREAMS

4.5 COMPUTER SYSTEMS SUPPORT

Basic support and graded evolution of the Illiac III Pattern Recognition Computer has been carried out under the present contract. This support has been central to the successful execution of the above core research program.

4.5.1 System Documentation

An extensive effort has been made to document the Illiac III Computer System. Key to this documentation has been the final issuance of the first two volumes of the Illiac III Reference Manual [5], [6]. The first of these includes a complete bibliography of available hardware documentation. Volumes III and IV have been prepared and will be issued this summer.

In engineering drawings, over 400 logical design sheets were prepared this past contract period in **final** form and signed off.

4.5.2 Design and Fabrication

The level of design work completed by the miniscule engineering (and maintenance) staff this past year is outstanding: 461 integrated circuit cards of control logic were laid out, and predominantly wired (under subcontract). A breakdown by area of the machine follows:

CONTROL CARDS OCTOBER 1 - JUNE 1

UNIT	# OF CARDS WIRED	# OF CARDS DESIGNED (NOT WIRED)	TOTAL
TP	73	54	127
PAU	73	2	75
SMV	81	31	112
AU	0	42	42
IU	0	30	30
SEQ. TEST.	25	7	32
PDP8/e	9	18	27
CIU	16	0	16
TOTAL	277	184	461

4.5.3 Control Point Strategy and Its Automated Diagnosis

The design technique known as Control Point Strategy has been used in the design of the Illiac III Pattern Recognition Computer. The control of this machine is functionally partitioned and physically divided into small, etched cards of integrated circuits. The testing of these cards is the immediate object of this research. Broader implications of this work are also discussed below.

4.5.3.1 Introduction

A control can be considered as being basically synchronous, asynchronous, pseudo-asynchronous or micro-programmed and the design techniques used are a function of these classifications.

The Department of Computer Science at the University of Illinois has been, historically, for the most part involved in the design and construction of asynchronous computers. The fact that this department has been primarily interested in parallel processor computers is strongly related to that involvement. An asynchronous control possesses some advantages over synchronous controls due to the relative independence of the speed of the different units. However, an asynchronous control "does require hardware which greatly increases the design complexity and renders a logical design problem of flow chart realization." Because of this complexity, this design has remained mostly "an art and not a science." (Swartout, 1963).

Some work [7], [8], [9], toward a formalization of this problem has previously been done and methods have been proposed to standardize the design. The solution proposed in [10] introduces a building block called a control point.

Work of the past year [10] has served to formalize and enlarge the control point concept. We introduce there a model called a control point net which yields a unique topological representation of a given pseudo-asynchronous system from which the control point design is easily derived.

4.5.3.2 Developments of the Past Year

A formal model of the pseudo-asynchronous control strategy has been developed [10]. Here a detailed analysis of the concepts of task, event, and transition in pseudo-asynchronous systems is given. A close study of what are called "external conditions" shows that transitions exhibit a very definite structure with respect to these conditions. This property is used to define a basic building block, called a control point, which logically implements the fundamental functions assumed by a transition.

The thesis of Rey [10] provides a brief review of how the control point strategy is actually realized in terms of logic circuits in the Illiac III Control. Given a control point design, a method based on the above formal model is then proposed for generating appropriate test patterns. The essential idea here is to partition the control point design into "independent subnetworks", to find tests for these subnetworks and to concatenate these tests according to a set of rules which take into account the connectivity properties of the Control Point Network. A first algorithm has been developed which realizes this concatenation.

The problem of physically applying the tests to individual cards was then attacked. The maintenance facilities provided by the Illiac III implementation of the control points, although simple, appear to be extremely time-saving for the debugging process. Since the actual test in our case is to be monitored by a mini-computer (PDP8/e), the design of a special interface, called the Sequence Tester, is needed. The Sequence Tester has two functions: (1) it submits input patterns to the card and (2) preprocesses the card response before transmitting it to the PDP8/e. The processing function facilitates the fault location.

4.5.3.3 Current Status

A detailed logical design of the Sequence Tester logic is essentially complete and fabrication has begun. The interface to the mini-computer (PDP8/e) should be largely complete before the end of the current contract year.

Test generation procedures have been implemented in a first FORTRAN program [11].

REFERENCES

[1] Read, John, "Show-and-Tell System Specifications," Department of Computer Science Report No. 429, University of Illinois, March 1971.

[2] Green, D. M., Swets, J. A., "Signal Detection Theory and Psychophysics," John Wiley & Sons, Inc., New York, 1966.

[3] Michalski, R. D., McCormick, B. H., "Interval Generalization of Switching Theory," Department of Computer Science Report No. 442, University of Illinois, May 3, 1971 (In printing).

[4] Schwebel, John C., "Picture Segmentation by Structural Inference," Thesis in preparation.

[5] "ILLIAC III REFERENCE MANUAL, VOL. I: The Computer System," edited by B. H. McCormick, and B. J. Nordmann, Jr., Department of Computer Science Report No. 433, February 17, 1971.

[6] "ILLIAC III REFERENCE MANUAL, VOL. II: Instruction Repertoire," edited by B. H. McCormick and B. J. Nordmann, Jr., Department of Computer Science Report No. 434, February 26, 1971.

[7] Petri, C. A. "Kommunikation Mit Automaten," Schriften des Reinish-West filischen Inst. Instrumental Math., and der Universitat Bonn, Nr. 2, Bonn 1962.

[8] Holt, A. W., Shapiro, R. M., Saint, H. and Warshall, S., "Final Report for the Information System Theory Project," Applied Data Research, Inc., ADR ref. 6608, February 1968.

[9] Patil, S. S., "Coordination of Asynchronous Events," M. I. T. MAC.TR, 72, June 1970.

[10] Rey, Christian A., "Control Point Strategy and Its Automated Diagnosis," Thesis work printed as Department of Computer Science Report No. 450, University of Illinois,, June 1971 (In printing).

[11] Michalski, R. D., Tareski, V. G., "Experiments with the Algorithm A^Q for the Interval Covering Synthesis, "Department of Computer Science Report in progress.

4.6 DOCUMENTATION

4.6.1 External Documents Issued

Report No. 442	Michalski, R. S., McCormick, B. H., "Interval Generalization of Switching Theory", May 3, 1971.
Report No. 450	Rey, Christian A., "Control Point Strategy and Its Automated Diagnosis", June 1971.
Report No. 461	Michalski, R. S. "A Geometrical Model for the Synthesis of Interval Covers," June 24, 1971. (In printing)
Report No. 463	Rey, Christian A., "Control Point Design Using Modular Logic," June 1971 (In printing)
File No. 862	Borovec, R. T., "IMAGE 8: PROGRAMMED I/O FACILITIES (The External Bus)," June 9, 1971.
File No. 863	Borovec, R. T., "IMAGE 8: The Real Time Clock", June 16, 1971.

Seminars: Sight Sensory Systems

"Intersection Detection and Sofa Problems," Kiyoshi Maruyama, Department of Computer Science, University of Illinois, April 14, 1971.

"The Pattern Articulation Unit of Illiac III: Past, Present, and Future," Richard T. Borovec, Department of Computer Science, University of Illinois, April 21, 1971.

"Speech Recognition Survey," Bernard J. Nordmann, Jr., Department of Computer Science, University of Illinois, May 5, 1971.

"Fault Diagnosis of Combinational Circuits and Its Application for the Check-out of Control Point Logic," Christian A. Rey, Department of Computer Science, University of Illinois, May 19, 197

"Prediction of Electronic Device Feature Behavior by Pattern Recognition Methods", Professor F. N. Pokrovsky, Visiting Professor at Purdue University from Moscow Power Engineering Institute, May 26, 1971.

4.6.2 Logic Drawings Issued

The following new logic drawings have been drawn during the past quarter:

Sequence Tester Logic	40 drawings
Image 8 (PDP8/e Interface) Logic	19 drawings
Transfer Memory II	46 drawings
T.P. Control Logic	82 drawings
A.U. Control Logic	5 drawings

The remaining A.U. Control Logic drawings will be processed in the immediate future.

4.6.3 Engineering Drafting Report

During the last quarter a total of 450 drawings, including new logic drawings, drawing changes, layouts, flow-charts, thesis, report drawings and drawings related to Opto/Mechanical design of Illiac III, have been processed by 2118 drafting section.

S. Zundo

4.7 ADMINISTRATION

4.7.1 Personnel Report

Senior Staff

Professor Bruce H. McCormick - Principal Investigator

Assistant Professor - R. S. Michalski

Professional Staff

Robert C. Amendola

Richard T. Borovec

John S. Read

Research Engineering Assistant

S. Paul Krabbe

Electronic Engineering Assistant

Joseph V. Wenta

Digital Computer Technician II

George T. Lewis

Drafting

Stanislavs Zundo

Secretarial

Mrs. Roberta Andre'

Research Assistants

Jerry Chen

Walter Donovan

Lakshmi Goyal

Richard P. Harms

S. N. Jayaramurthy

Ahmad Masumi

Kiyoshi Maruyama

Bernard J. Nordmann

Peter Raulefs

Christian Rey

John C. Schwebel

Val Tareski

Kuo Wen

4.7.2 Computer Usage Log Summaries

1. The PDP8/e, which arrived early in June, was used approximately 100 hours, mostly for installation and checkout of interfaces to the Illiac III and IBM 360/75.

2. The Scanner-Monitor-Video System was used as follows:

Production & demonstration	223 hours
Preventive maintenance	16 hours
Corrective maintenance	12 hours

3. IBM 360/75 utilization totaled $3,718.12

J. Read

Form AEC-427
(6/68)
AECM 3201

U.S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.
COO-2118-0015

2. TITLE
QUARTERLY TECHNICAL PROGRESS REPORT - APRIL-JUNE, 1971

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report
□ b. Conference paper not to be published in a journal:
Title of conference ______
Date of conference ______
Exact location of conference ______
Sponsoring organization ______
□ c. Other (Specify) ______

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.
□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)
Professor Bruce H. McCormick
Principal Investigator
Illiac III Project

Organization
Department of Computer Science
University of Illinois
Urbana, Illinois

Signature Bruce H. McCormick

Date July 7, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.
□ b. Report has been sent to responsible AEC patent group for clearance.
□ c. Patent clearance not required.

5. ILLIAC IV

(This work was supported in part by the Department of Computer Science, University of Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.)

REPORT SUMMARY

During the quarter the majority of the efforts in off-line diagnostics were concentrated in conversion of the card test generation system from the B5500 to the B6500. Automation Technology, Inc. maintenance engineers have been actively participating with Burroughs personnel in the on-line debugging of ILLIAC IV.

Work on a new technique to allow easier test generation through use of Roth's D-Algorithm has progressed well. Work next quarter will concentrate on larger logic partitions.

Progress has been made in the ILLIAC IV FORTRAN and operating system. Despite manpower shortages and an erratic B6500, all languages have been brought to what is essentially a refining stage. The operating system is on the verge of complete integration where it can be used to completely simulate the present ILLIAC IV system. Documentation of software has been extended and refined as evidenced by a library of macros and subroutines.

Two hardware interfaces between the PDP-11 and the ARPA network IMP were completed and debugged. Development of the ARPA Network Terminal System (ANTS) proceeded with near completion of the software coding, and staffing personnel for a graphics effort was begun.

Application efforts have continued in solving partial differential equations, solutions of systems of linear equations, eigenvalues, polynomial root finding and time series analysis. Education efforts included two seminars held during the quarter.

Project expenditures through June, 1971:

Burroughs Corporation	$27,477,000.00
University of Illinois	7,486,608.87

HARDWARE

5.1 Off-Line Diagnostics

5.1.1 PE Test

PE Control Logic Tests are as yet incomplete. Manpower originally scheduled to complete this area during the quarter was diverted to program conversion.

5.1.2 Card Test

Work on Card Test Generation was halted during the quarter while program conversion to the B6500 was taking place. Work is expected to resume in this area during the next quarter.

5.2 Program Conversion - ATI

The majority of the efforts in off-line diagnostics were concentrated in conversion of the card test generation system from the B5500 to the B6500. The problems of poor reliability and poor or misleading documentation, common to other areas of the project, had an impact in this area also. However, operation of the B6500 version of the system is expected during the next quarter.

5.3 Development - ATI

Work on a new technique to allow easier test generation through use of Roth's D-Algorithm has progressed well. Initial programming is complete and has been tested using PE cards. Work next quarter will concentrate on larger logic partitions, such as the CU cards and how to make them operationally available as a new aid in test generation.

5.4 ILLIAC IV Maintenance

ATI maintenance engineers have been actively participating with Burroughs personnel in the on-line debugging of ILLIAC IV. The major areas which had active participation during this period were: PE debugging and modification, checkout of repaired PE/MLU cards, CU debugging, modification to spare cards for the CU, and the complete checkout/ modification to the I/O area. The specification for a confidence test has been prepared, and the program is over 50% completed. Also the spare part break out needed for the operation of ILLIAC IV has been completed and supplied to the University.

5.5 Financial

At this point in the contract, ATI is approximately seven percent (7%) below the budgeted cost estimate.

5.6 Operating System

This quarter the major effort of the operating system group has been directed toward integrating, instrumenting, and stressing those parts of the operating system that reside in the B6500. The operating system has been integrated with the ILLIAC IV simulator to the extent that the Job Partner may communicate with a simulation of OS4 through a simulated TMU. Work presses forward on a broad band data path analogous to that of the BIOM and ILLIAC IV disk. A full simulation of the working of the I/O subsystem will not be attempted, however.

The operating system has been made "half-load proof"; i.e., it has been made largely impervious to B6500 failures. This is done by storing vital data and program status on the B6500 disk so that the operating system is almost always restartable.

The instrumentation added to the operating system this quarter allows the system to log the amount of CPU and I/O time spent in each operating system module by each ILLIAC IV job. It aids in allowing detailed job histories to be produced as a listing for the benefit of the user and of those maintaining the system. It allows the computer operators to interrogate the operating system to determine the status of any particular ILLIAC IV job, and lets the operator manipulate the operating system in case of hardware or software failure.

The ILLIAC IV resident module of the operating system, OS4, is being rewritten to take advantage of hardware interrupt changes and more flexible I/O capabilities.

Preliminary talks have been held with Burroughs Corp., Paoli, to discuss the integration with the On-line Diagnostic Programs (IDIAP) of the operating system.

Work has also begun on a special set of utility programs to effect format conversion between the various B6500 binary word formats and those of ILLIAC IV. These so-called "symbionts" each have a special Job Partner, and execute in the reserved area of the array. They do not necessarily have any relationship with the current ILLIAC IV job and are, in reality, part of the data processor which currently executes on the B6500 only. By transferring this array oriented module to ILLIAC IV a considerable load will be removed from the B6500.

By the end of next quarter a preliminary version of the ILLIAC IV operating system will be ready to run as soon as the hardware debugging schedules permit.

5.7 Compilers and Translators

5.7.1 Cockroach

A working version of Cockroach (ILLIAC IV FORTRAN) is now available in the form of a Cockroach-to-Glypnir translator. This version includes functions, subroutines, and arrays of up to 64 in the "row" dimension. It has been successfully used to write and simulate programs by non-University users.

During the last two months of the quarter, the Cockroach group's efforts have been directed toward a quantitative study of the storage and execution efficiency of various storage allocation schemes. These schemes are for handling arrays whose row dimension is greater than the 64 which have been proposed in the past. The storage scheme is due to be completed this quarter.

Additional documentation for this language is being completed.

5.7.2 Glypnir

Some difficulty was experienced in transferring the Glypnir compile from the B5500 to the B6500. This was largely due to differences in MCP intrinsics between the two machines and their documentation.

Most of the quarter was devoted to reorganizing the compiler in preparation for including facilities which would allow the binding of separately compiled Glypnir subroutines with a Glypnir program. Effort was also spent in "cleaning up" the present compiler to make it faster and more reliable. Various operations of the compiler were timed and studied with a view to making a major speedup at a later date.

5.8 Assembler (ASK)

The Assembler has undergone a major revision during the quarter. Its peak speed has been more than doubled to over 2000 cards per minute; its output listing has been made more legible and can be printed faster, and a lot of dead wood has been cut from the compiler itself. In addition, a considerable number of "bugs" have been fixed. No changes have been made in the language.

5.9 Simulator (SSK)

Work on the simulator was concluded during this quarter. After transferring it to the B6500, a confidence test was written to test nearly all SSK functions. This is now in use. The simulator is tested for correct functioning each time there is a change of Burroughs software. No more work will be done on the simulator unless absolutely necessary.

5.10 Interactive Communications and Graphics

5.10.1 Interactive Communications

During the reporting period, two hardware interfaces between the PDP-11 and the ARPA network IMP were completed and debugged. The first interface will remain at the University of Illinois terminal site. The second interface was sent to Paoli, Pennsylvania. The second PDP-11 system and appropriate interfaces for connection to the B6500 at Paoli were checked out and delivered to Paoli for installation during the next reporting period.

Additional equipment received during this period included: Gould Electrostatic Printer/Plotter and a Computek storage scope, Model 400/15, terminal system. These two devices were successfully interfaced to the PDP-11 and programming efforts for support packages were started.

Development of the ARPA Network Terminal System (ANTS) proceeded with near completion of the software coding. Initial debug and checkout operations on the network NCP were also started. Subsidiary programming efforts were started to develop a version of ANTS for the Paoli PDP-11 (PANTS) to allow a specialized remote access link between the University and the Paoli B6500. Completion of this part of the project is scheduled for August, 1971.

5.10.2 Graphics

During the quarter, staffing of personnel for a graphics effort was begun. A level of 2 1/2 full-time employees was reached, and efforts were started to provide for implementation of support packages for graphics devices such as the Gould printer/plotter, Computek storage scope, network access to graphics utilities sites such as Utah, BBN, MIT, etc. In addition, work was started on a basic graphics structure system to provide a generalized support structure linking all Center graphics activities and activities on the ARPA network.

5.11 Library

A small library of ILLIAC IV subroutines and macros, together with a program to aid the general user in collecting them in proper sequence, has been made available. These include SIN, COS, ARCTAN, LN, SQRT, and EXP in 32- and 64-bit mode. ITU routines and macros are also available. However, more work needs to be done before these can be considered truly "professional."

APPLICATIONS

5.12 Numerical Analysis

5.12.1 Partial Differential Equations

5.12.1.1 Block Jacobi Method

Work on the Block Jacobi method is being continued to solve the elliptic type partial differential equations for given boundary conditions.

Let R be a rectangular region with boundary S, and let g(x) be a continuous function defined on S. Find a function u(x,y) which is continuous on R + S, satisfies on R the following equation:

$$a(x,y) \frac{\partial^2}{\partial x^2} u(x,y) + b(x,y) \frac{\partial^2}{\partial y^2} u(x,y) + c(x,y) \frac{\partial}{\partial x} u(x,y)$$

$$+ d(x,y) \frac{\partial}{\partial y} u(x,y) + e(x,y) u(x,y) = f(x,y) \qquad (1)$$

and satisfies on S the condition u(x,y) = g(x,y) (Dirichlet condition). Sometimes $\frac{\partial u}{\partial \eta}$ $(\frac{\partial}{\partial \eta}$ means the normal derivative--Neumann condition) or linear combination of u and $\frac{\partial u}{\partial \eta}$ is prescribed on S instead of u itself. The following are analytic functions of X and Y in R and satisfy in R + S the conditions $a > 0$, $b > 0$ and $e < 0$: a, b, c, d, e and f.

The corresponding finite difference equations on n x n mesh points in R are given by,

$$c_{1,i,j} u_{i-1,j} + c_{2,i,j} u_{i+1,j} + c_{3,i,j} u_{i,j-1} + c_{4,i,j} u_{i,j+1}$$

$$+ c_{5,i,j} u_{i,j} + c_{6,i,j} = 0 \qquad 1 \leq i,j \leq n \text{ (= even number)} \qquad (2)$$

where i, j are indices of mesh points.

In point-iterative methods the value of each component $u_{ij}^{(n)}$ is determined by an explicit linear formula. That is, the nth approximation of the (i,j) component can be expressed by using already computed values of the other unknowns. In block-iterative methods, on the other hand, groups of components of $u^{(n)}$ are modified simultaneously by solving a system of linear equations. Therefore individual components are defined implicitly in terms of the others in the same group.

The Block Jacobi method is one of the block-iterative methods and is equivalent to the successive line over the relaxation method (SLOR) except that the acceleration parameter is changed at each step in the former case.

We write the system of equations (2) in matrix notation as

$$\begin{bmatrix} D_1 & F_1 & & & \\ E_1 & D_2 & F_2 & & \\ & E_2 & D_3 & F_3 & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ \vdots \\ \vdots \end{bmatrix} \qquad (3)$$

or in short $A_u = k$ (3')

where D_i's are tridiagonal matrices, E_i's and F_i's are diagonal matrices, $u'_i = (u_{i1}, u_{i2}, \ldots u_{in})$ and $k'_i = (k_{i1}, k_{i2}, \ldots k_{in})$ which consists of boundary values and $C_{6,i,j}$'s, (Components of D, E, F consist of C_m's $(m = 1, 2, \ldots, 5)$).

If we interchange the order of components of u in (3') so that the mesh points with odd i's are in the upper half of u and those with even i's are in the lower half of u without changing j's (in other words, to separate odd rows from even rows), we have the following system of linear equations instead of (3):

$$\left[\begin{array}{cccc|cccc}
D_1 & & & 0 & F_1 & & & 0 \\
 & D_3 & & & E_2 & F_3 & & \\
 & & D_5 & & & E_4 & F_5 & \\
0 & & & \ddots & 0 & & E_6 & \ddots \\ \hline
E_1 & F_2 & & 0 & D_2 & & & 0 \\
 & E_3 & F_4 & & & D_4 & & \\
 & & E_5 & F_6 & & & D_6 & \\
 & 0 & & \ddots & 0 & & & \ddots
\end{array}\right] \cdot \left[\begin{array}{c} u_1 \\ u_3 \\ u_5 \\ \vdots \\ \hline u_2 \\ u_4 \\ u_6 \\ \vdots \end{array}\right] = \left[\begin{array}{c} k_1 \\ k_3 \\ k_5 \\ \vdots \\ \hline k_2 \\ k_4 \\ k_6 \\ \vdots \end{array}\right] \tag{4}$$

In this new scheme, the (m+1)st iteration is performed in the following way:

1) Solve

$$\begin{bmatrix} D_1 & & & \\ & D_3 & & \\ & & D_5 & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \\ u_{n-1} \end{bmatrix}^* = \begin{bmatrix} k_1 \\ k_3 \\ k_5 \\ \vdots \\ k_{n-1} \end{bmatrix} - \begin{bmatrix} F_1 & & & \\ E_2 & F_3 & & \\ & E_3 & F_5 & \\ & & E_6 & \ddots \end{bmatrix} \begin{bmatrix} u_2 \\ u_4 \\ u_6 \\ \vdots \\ u_n \end{bmatrix} \tag{5}$$

for $\begin{bmatrix} u_1 \\ u_3 \\ \vdots \\ u_{n-1} \end{bmatrix}^*$. Then obtain $\begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m+1)}$ by extrapolation

$$\begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m+1)} = \begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^* + \alpha_m \begin{bmatrix} u_1^* - u_1^{(m)} \\ u_3^* - u_3^{(m)} \\ u_5^* - u_5^{(m)} \\ \vdots \end{bmatrix} \tag{6}$$

where α_m is the mth acceleration parameter for odd lines.

2) Solve

$$\begin{bmatrix} D_2 & & & 0 \\ & D_4 & & \\ & & D_6 & \\ 0 & & & \ddots \end{bmatrix} \begin{bmatrix} u_2 \\ u_4 \\ u_6 \\ \vdots \\ u_n \end{bmatrix}^{*} = \begin{bmatrix} k_2 \\ k_4 \\ k_6 \\ \vdots \end{bmatrix} - \begin{bmatrix} E_1 & F_2 & & & 0 \\ & E_3 & F_4 & & \\ & & E_5 & F_6 & \\ 0 & & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m+1)} \qquad (7)$$

for $\begin{bmatrix} u_2 \\ u_4 \\ u_6 \\ \vdots \end{bmatrix}^{*}$. Note that $\begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m+1)}$ is used instead

of $\begin{bmatrix} u_1 \\ u_3 \\ u_5 \\ \vdots \end{bmatrix}^{(m)}$ on the right hand side of (7).

Then obtain the (m+1)st approximation by extrapolation

$$\begin{bmatrix} u_2 \\ u_4 \\ u_6 \\ u_8 \\ \vdots \end{bmatrix}^{(m+1)} = \begin{bmatrix} u_2 \\ u_4 \\ \vdots \end{bmatrix}^{*} + \alpha_{m+1/2} \begin{bmatrix} u_2^{*} - u_2^{(m)} \\ u_4^{*} - u_4^{(m)} \\ \vdots \end{bmatrix}$$

where $\alpha_{m+1/2}$ is the mth acceleration parameter for even lines.

α_i's are given either in the form of a table or calculated recursively from the previous value at each step ($\alpha_m = f(\alpha_{m-1/2})$).

Note that in either step a system of linear equations for each row can be solved independently of others. This is a very nice feature for parallel computation; one PE solves one row. The usual Thomas algorithm is used to solve linear equations of tridiagonal matrices.

Each PE contains two adjacent rows so that all 64 PE's are enabled when N = 128. The original version in which each PE contains one row has been implemented in Glypnir and has been run successfully. The program for the new storage scheme is being programmed and debugged.

5.12.1.2 Numerical Solutions of Problems in Hydrodynamics

A computer program using the Lax-Wendroff finite difference scheme, which has been tested with Burger's Equation [1], is under development for calculating the compressible, inviscid, transonic flow around a circular cylinder. A consistent formulation of the problem, which is eminently suitable for parallel computation, has been successfully arrived at. This is done in such a manner that, for shockless flow, no additional problems are posed in satisfying the boundary conditions at the body surface as well as at the far field free stream boundary. The same instructional stream is fed to each arithmetic unit whether it is computing boundary point values or interior point values. Thus no computing time is lost at the boundary points. This leads to high machine efficiency. In addition, a computational algorithm that permits a change in the mesh sizes with a minimum of change in the instructional stream to the processing units is under investigation.

The computational technique is based on the asymptotic method of calculating steady state solutions using the time dependent formulation [2]. The two-dimensional problem is computed in cylindrical polar coordinates. Variable mesh sizes are employed to accommodate the rapid variation of the physical parameter, especially at the forward stagnation point.

Preliminary results from the computer program indicate that the formulation currently employed does yield a solution which exhibits properties similar to the solution for inviscid flow obtained by Moretti [3]. The present formulation, however, possesses the added advantage of being amenable to extension to three dimensions and more complicated body shapes for practical applications.

5.12.2 Solutions of Systems of Linear Equations

During this quarter, the ALGOL versions of three subroutines were converted to B6500 ALGOL, debugged, and documented. They are now available for use. The three routines are:

(1) Triangular decomposition of a square matrix

(2) Gaussian elimination

(3) Matrix inversion (triangular decomposition method).

The ASK versions of the above 3 routines are partially debugged. Changes are being made currently in the storage allocation scheme and I/O scheme. When these changes are completed, the routines will be able to handle matrices up to 1500 x 1500 (as opposed to 630 x 630), and the I/O wait time will be significantly reduced in most cases. Debugging will continue as soon as these changes are completed. A subroutine to perform the conjugate gradient method has also been started.

We have devoted ourselves to two main activities concerning solution of systems of linear equations and matrix inversion. First, B6500 ALGOL subroutines, namely HOUSEHOLDER and MODIFY, have been coded, debugged, and tested on the machine. A double precision version of HOUSEHOLDER has been implemented to test several ill-conditioned matrices. The second effort has been to document existing programs according to ILLIAC IV subroutine standards. The documents are available for the ALGOL programs, and the ASK documents are being prepared. The HOUSEHOLDER upper triangularization method as described in QPR 1970 [4] has been modified for purposes of decomposing an n x n matrix A of rank r into the product of two matrices A = BC where B is an n x r matrix and C is an r x n matrix.

5.12.3 Eigenvalues

5.12.3.1 The Eigenvalue Problem

In this quarter, the following three ALGOL programs have been coded and tested successfully:

(1) Solution of systems of complex linear equations (CU decomposition)

(2) Inverse iteration for finding eigenvectors

(3) Eigenvector solution of the matrix Riccati equation.

Actually, programs (1) and (2) are a part of program (3), but for the purpose of completeness and for their independent utility, they are tested separately with numerical examples. These will be documented in ALGOL and GLYPNIR in the next quarter.

For problem (3), both Eberlein's and QR's methods [5] are used, depending on whether the matrix of coefficients of the corresponding linear dynamic system is diagonable.

Two methods of transforming a matrix into Upper Hessenberg form,

a) using elementary unitary transformations (HOUSEHOLDER's method)

b) using elementary stabilized transformations,

were investigated. It was concluded that the second method is efficient only for core-contained matrices and has been coded in B6500 ALGOL (File MMTI/UH/ALG) (debugged and running) and in ASK (MMTI/UH/I4ASK) (being debugged). For non-core contained matrices, in order to keep disk memory transfer minimum, the HOUSEHOLDER's method is being coded in ASK. Macro definitions for using the ILLIAC IV disk as a direct-access device are being coded to facilitate access to rows and columns stored on disk and their transfer to core memory. The main problem is to have the double QR-algorithm for non-core contained matrices running in ILLIAC IV, taking advantage of the parallelism of the machine by choosing an appropriate value of the origin shift. For this purpose the algorithm has been coded in B6500 ALGOL (MMTI/QR/ASG) (debugged and running) to experiment with different matrices, and if possible, empirically determine the best origin shift and the suitability of this choice for the non-core contained matrices and the storage scheme both in ILLIAC IV disk and memory.

For the purpose of numerical stability, the matrices have to be balanced. The method described by Parlett and Reinsch in their paper "Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors" is to be coded as a subroutine which produces the balanced matrix.

5.12.3.2 Eigenvalues and Eigenvectors of Symmetric Tridiagonal Matrices

5.12.3.2.1 QR-algorithm for Symmetric Tridiagonal Matrices

QR-algorithm for finding the eigenvalues of symmetric tridiagonal matrices written last quarter has been modified into an independently compiled procedure on the B6500. The procedure has been debugged and tested for different-size matrices. The calling sentence is

STREVA (D,E,N,LMT,EVA) ; .

The complete document of this procedure has also been completed. This algorithm is now being written for the ILLIAC IV in GLYPNIR.

5.12.3.2.2 Inverse Iteration for the Corresponding Eigenvectors

This algorithm written last quarter has been modified into an independently compiled procedure on the B6500, debugged, and tested. The calling sentence is STREVC (C,B,N,W,M1,MACHEPS,Z) ; . The complete document of this procedure has also been finished. This algorithm is now being written for the ILLIAC IV in GLYPNIR.

5.12.3.2.3 Iterative Method for Solution of the Eigenproblem

Gradient methods for finding the maximal (or minimal) eigenvalue and its corresponding eigenvector for large scale symmetric matrices are being explored for the ILLIAC IV. The algorithm can be described as follows.

Consider the Rayleigh quotient as a function f of n variables $\bar{x}$, i.e.,

$$f(\bar{x}) = (A\bar{x}, \bar{x})/(\bar{x}, \bar{x})$$

then the gradient

$$\nabla f = g = 2(A\bar{x} - f\bar{x})/(\bar{x}, \bar{x})$$

can be readily evaluated. Maximize (or minimize) the function f by gradient methods, we get the maximal (or minimal) eigenvalue of matrix A, and the final value of vector $\bar{x}$ is its corresponding eigenvector.

5.12.4 Polynomial Root Finding

Two polynomial root-finding routines have been coded, debugged and documented for B6500 ALGOL. The first is a procedure finding real roots by using Sturm sequences. The second finds all roots, real and complex, using root squaring and sub-resultant procedures.

A real root finder has been coded in ASK and will be debugged. A document on this routine will come out August 11, 1971.

5.12.5 Time Series Analysis

5.12.5.1 Identification and Estimation of Stochastic Models

Algorithms for identifying Stochastic models and estimating their parameters have been studied in this period. Programs are being written in ALGOL. Comparisons are to be made among several methods, such as the maximum likelihood method, the linear least-squares method and other non-linear methods, to estimate the parameters of the models.

5.13 Linear Programming

The past quarter has again seen an assembler-simulator package insufficient to facilitate code debugging. Each attempted simulation primarily served to assist in SSK debugging. With this in mind, a brief, intensive joint effort between LP and the ILLIAC IV Project's simulator maintenance group was undertaken to produce as SSK exerciser. While the work was considered a success, at quarter's end the simulator remained nearly unusable.

Unable to proceed with INVERT debugging, attention returned to the SETUP and SETDOWN codes--those programs massaging data from "external" storage conventions into the format of the ILLIAC IV LP package, and back again. Over the quarter, the early B5500 routines were translated from XALGOL to ALGOL. Major recoding was done to accommodate a workload shift from the B6700 to the ILLIAC IV. As the period closed preliminary flowcharts were being developed to extend the ILLIAC IV's load to include label identification and data allocation--the primary work remaining on the B6700.

Major SETDOWN codes were written and syntactically debugged, with synthetic test-data being developed. In developing the SETDOWN procedures, some output formats were modified to facilitate translation of solution information. These modifications are being incorporated into the LPS code and documentation. It now appears quite possible for virtually the entire SETUP/SETDOWN package to be placed on the ILLIAC IV, with the primary constraint to the realization of this being time and manpower available.

Some time has been spent this quarter exploring the possibility of a collaborative effort with the group of Professor A. Charnes at University of Texas. Particular areas of mutual interest may be algorithm development for mathematical programming and the solution of large linear programming models. This effort will continue into the next quarter.

5.14 Signal Processing

The following ALGOL procedures have been documented and tested for use on the B6500 computer:

(1) a plotting procedure called GRAPH,

(2) a procedure called FHANN for computing the impulse response and power of a band-pass filter,

(3) a one-dimensional fast Fourier transform procedure called FFT,

(4) a one-dimensional inverse fast Fourier transform procedure called IFT,

(5) a one-dimensional conventional Fourier transform procedure called FURY,

(6) an auto correlation procedure called AUTO.

5.15 Education

5.15.1 Teaching

During this quarter, activities included teaching the graduate course on ILLIAC IV for the University of Illinois, a two-day ILLIAC IV seminar for Lawrence Radiation Laboratories (LRL) and a one-day seminar for attendees of the Fluid Dynamics Workshop given by the staffs of the Center for Advanced Computation, Coordinated Science Laboratory and ILLIAC IV Project at the University of Illinois.

5.15.1.1 CS-491: Architecture, Applications and Languages for a Parallel Computer

The Spring Semester of CS-491 had 14 students; 4 took the course for one unit, 5 for 1/2 unit and 5 audited the course. The course material covered was substantially the same as in previous semesters (refer to outline in previous QPRs) but with slightly more emphasis on the ILLIAC IV programming languages--ASK, GLYPNIR, and FORTRAN.

The content next fall will stress more heavily the operation of the ARPA Network as ILLINOIS becomes more and more a working node of the Network.

In order to make CS-491 accessible to a larger class of students at the University of Illinois, the course will be changed to the 300 level (can be taken for graduate credit or by advanced undergraduates). The new course designation will be CS-397: Special Topics in Computer Science--The ILLIAC IV Computer System. Following is a course description:

CS 397 Special Topics in Computer Science--
The ILLIAC IV Computer System

Review of conventional and unconventional organization hardware structure, diagnostic and test/repair equipment, programming languages, operating system and selected applications. ILLIAC IV will be accessed remotely in the solution of class problems. One-half to one unit. Prerequisite: Consent of instructor.

5.15.1.2 Two-Day Seminar for LRL

On May 13 and 14 a two-day seminar on ILLIAC IV was given to three employees of LRL. The seminar outline follows:

May 13, 1971

I. Overview 10:00 am - 12 noon

- A. Background Conventional and Unconventional Organizations
  - 1. Overlap
    - a. Buffer
    - b. Pipeline
  - 2. Replication
    - a. General Multiprocessor
    - b. Recentralize Memory
    - c. Recentralize ALU - Intrinsic Multiprocessor
    - d. Recentralize CU - Vector or Array Processor
  - 3. Both 1) and 2) above
- B. ILLIAC IV is a Vector Processor
- C. Hardware Structure
  - 1. Organization Chart
  - 2. ILLIAC IV Array - General Description
  - 3. Some Illustrative Problems
  - 4. ILLIAC IV I/O System
  - 5. ILLIAC IV System
  - 6. Some Indicators of Powers of ILLIAC IV
- D. Diagnostics and Test/Repair Equipment

E. Software

F. Applications

II. Programming Languages 2:00 pm - 5:00 pm

A. ASK

1. Background - Review, Notation, Conventions

2. Sample Problems

a. Summing an Array of Numbers

b. Finding the Largest Value in an Array of Numbers

<u>May 14, 1971</u>

B. ALGOL 1:00 pm - 1:30 pm

C. GLYPNIR 1:30 pm - 3:00 pm

5.15.1.3 One-Day Seminar for Fluid Dynamics Workshop

On June 15 a one-day seminar on ILLIAC IV was given to 26 attendees of the Fluid Dynamics Workshop. Following is the seminar outline:

I. Overview 9:00 am - 12 noon

A. Background Conventional and Unconventional Organizations

1. Overlap

a. Buffer

b. Pipeline

2. Replication

a. General Multiprocessor

b. Recentralize Memory

c. Recentralize ALU - Intrinsic Multiprocessor

d. Recentralize CU - Vector or Array Processor

B. ILLIAC IV is a Vector Processor

C. Hardware Structure

1. Organization Chart

2. ILLIAC IV Array - General Description

3. Some Illustrative Problems

4. ILLIAC IV I/O System

5. ILLIAC IV System

6. Some Indicators of Powers of ILLIAC IV

D. Diagnostics and Test/Repair Equipment

E. Software

F. Applications

G. Current Status

II. Programming Languages 2:00 pm - 5:00 pm

A. ASK

1. Background - Review, Notation, Conventions

2. Sample Problems

a. Summing an Array of Numbers

b. Finding the Largest Value in an Array of Numbers

5.15.2 Consulting

A small amount of time was spent this quarter in one-to-one consulting with people from the Center for Advanced Computation, ILLIAC IV, and people outside the University of Illinois concerning the ILLIAC IV and B6500.

5.15.3 Training

Education has taken on the responsibility of training two Affirmative Action employees to become Computer Programmers. In order to teach the trainees, a master-apprentice program is being used--the trainee meets with a different experienced programmer for one hour each day. Each experienced programmer teaches the trainee as much of the crafts of programming as he feels the trainee needs at his current level of development. The method of teaching ranges from lectures, to assigning problems, to working directly with the trainee.

Additionally, the trainees are learning the BASIC language for the PDP-11 and ALGOL for the B6500 and are solving a prescribed set of problems designed to instill the fundamentals of algorithmic problem-solving. Each problem is flow-charted, coded in both BASIC and ALGOL and finally run on the appropriate machine to solution.

5.15.4 Technical Writing

5.15.4.1 An Introductory Description of the ILLIAC IV System: ILLIAC IV Document No. 225

The first version of ILLIAC IV Document 225 will be ready for distribution early next quarter. Following is ing instructions for the document:

Abstract

Written specifically for an applications programmer, the book presents a tutorial description of the ILLIAC IV System. The initial version contains three chapters--Background, Hardware Structure, and The Assembly Language--ASK. Many illustrative problems are used to educate the beginner in the use of the ILLIAC IV System.

Reading Instructions

This book was written for an applications programmer who would like a tutorial description of the ILLIAC IV system before attempting to read the reference manual. As a tutorial, the level of detail presented in this book is fairly general; particular information can be found in the Burroughs Reference Manual "ILLIAC IV Systems Characteristics and Programming Manual."

In order to use this book most effectively, the chapters should be read in order. The reader who wants a very quick look at the capabilities of ILLIAC IV can skim just the summaries of parts A, B and C of Chapter I and begin reading on page II-55. He may then read pages II-1 through II-19, skipping the detailed description of the ILLIAC IV Array. Pages II-40 through II-72 are optional; the reader should at least look at them and decide for himself. As much of Chapter III as possible should be read--the instruction repertoire, more than anything else, defines the capabilities of a computer. A valid answer to the question "What is ILLIAC IV?" would be to hand the questioner a description of each instruction in the repertoire.

For a more complete understanding, however, the reader should come back and read the sections he skipped on the first pass. It is the nature of ILLIAC IV, to a degree much greater than the conventional computers, that its hardware structure is bound up very closely with its capabilities. It is therefore necessary that the reader spend the time necessary to understand the architecture of ILLIAC IV.

The Table of Contents in the front of the book is in an abbreviated format while each chapter will be preceded by a finer Table of Contents. Each chapter will also have its own Glossary which can be detached for handy reference in addition to a cumulative Glossary at the end of the book. As these become available they will be distributed to you.

Chapter I presents the background concepts necessary for an understanding of ILLIAC IV. A short section is devoted to the historical development of digital computers, and their evolution is described in terms of the problems that had to be solved. After conventional computer organizations are described, unconventional ones are presented as design options to speed up the operation of a computer. Two design philosophies, overlap and replication, represent two major methods used to increase the computer's operational speed. Overlap is effected by the buffer and pipeline mechanism and replication is embodied in the general multiprocessor. ILLIAC IV is shown to be a variant of a general multiprocessor using buffering and a modified pipeline mechanism in the instruction execution section.

Chapter II describes the architecture or the hardware structure of ILLIAC IV. The ILLIAC IV Array is discussed in broad terms followed by some illustrative problems which point out some of the similarities and differences between problem-solving on sequential and parallel machines. The problems also serve to illustrate how the hardware components are tied together. Following is a more detailed description of the ILLIAC IV Array, then another illustrative problem (Laplace's equation describing steady-state temperature distribution in two-dimensions) followed by some data allocation considerations; the ILLIAC IV I/O System is discussed briefly, and some conclusions and opinions end the chapter.

Chapter III presents the Assembly Language ASK in a functional and pragmatic way: a problem is described and then only those ASK instructions necessary for the solution are described. In this way the five problems introduce forty ASK instructions and the flavor of the assembly language which, from a programmer's standpoint, is an indication of the capabilities of ILLIAC IV itself. The five problems are: Summing an array of numbers, Finding the largest value in an array of numbers, Matrix multiplication, Matrix transpose, and Laplace's equation described in Chapter II.

Chapters IV through XI and the Glossary are not included in this edition but will be supplied as soon as they become available.

5.15.4.2 IEEE Article

Education is coordinating and contributing to a future article to be presented in the IEEE Proceedings under the tentative title "The ILLIAC IV Supercomputer." The first draft will be sent to the IEEE by the end of the next quarter.

ADMINISTRATION

5.16 Administration and Services

5.16.1 Contract Status

Burroughs has advised that the schedule for completion of the ILLIAC IV Computer has slipped from June to August and September.

Item	From	To
Diagnostic Programs	June 1971	September 1971
Deliver One Quadrant ILLIAC IV System. . . .	June 1971	September 1971
Commence Final Acceptance Testing of ILLIAC IV	June 1971	August 1971
Documentation	June 1971	September 1971
Engineering Drawings	June 1971	September 1971
Test Equipment	June 1971	September 1971

5.16.2 Financial Report

Budgeted expenditures--Fourth Quarter, Fiscal Year 1971

	April	May	June	Total for Quarter
Burroughs	$136,234.00	$136,295.00	-0-	$272,529.00
University	215,302.00	215,302.00	$215,302.00	645,906.00

Actual expenditures and commitments--Fourth Quarter, Fiscal Year 1971 (through May, 1971)

	April	May	June	Total for Quarter
Burroughs	$199,000.00	$175,000.00	*-0-	$374,000.00
University	330,810.19	341,524.85	**-0-	672,335.00

Total budgeted and actual expenditures to date:

	Total Budgeted Expenditures Through June, 1971	Total Actual Expenditures Through May, 1971
Burroughs Corporation	$27,687,946.00	$27,477,000.00
University of Illinois	7,673,203.92	7,486,608.00

*NA - Report of Expenditures from Burroughs covering June not received as of July 6, 1971.

** - Report of University expenditures for June not completed as of July 6, 1971.

REFERENCES

[1] Rajan, S., "Numerical Solution of the Partial Differential Equations of Gas Dynamics," In preparation, Center for Advanced Computation, University of Illinois at Urbana-Champaign.

[2] Magnus, R., and Yoshihara, H., "Inviscid Transonic Flow Over Airfoils," A.I.A.A. Journal, Dec. 1970.

[3] Moretti, G., "Transient and Asymptotically Steady Flow of an Inviscid, Compressible Gas Past a Circular Cylinder," Polytechnic Inst. of Brooklyn Report.

[4] ILLIAC IV Quarterly Progress Report, July, August, and September, 1970. "Matrix Inversion and Solution of Linear Algebraic Equations." ILLIAC IV Document No. 236, DCS Report No. 415 (October 15, 1970).

[5] ILLIAC IV Quarterly Progress Report, July, August and September, 1970. "The Eigenvalue Problem." ILLIAC IV Document No. 236, DCS Report No. 415 (October 15, 1970).

THESES

Alsberg, P.A. "OSL/2 An Operating System Language." Ph.D. Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

Beals, A.J. "The Automatic Generation of Deterministic Parsing Algorithms." Ph.D. Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

LaFrance, J.E. "Syntax-Directed Error Recovery for Compilers." Ph.D. Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

Nakamoto, H. "On-Line Diagnosis of ILLIAC IV." Master's Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

DOCUMENTS

Gary, J. "PDELAN--A Programming Language for the Solution of Partial Differential Equations." ILLIAC IV Document No. 229, Department of Computer Science File No. 853, ILLIAC IV Project, University of Illinois at Urbana-Champaign (July 31, 1970).

Koga, Y., K. Naemura and C. Tanaka. "Logic Simulation for the ILLIAC IV Processing Element." ILLIAC IV Document No. 223, Department of Computer Science File No. 849, ILLIAC IV Project, University of Illinois at Urbana-Champaign (January 22, 1971).

Schuster, S.A. "The Tuning of Buffer Parameters for the ILLIAC IV Data Management System." CAC Document No. 3, Center for Advanced Computation, University of Illinois at Urbana-Champaign (January 15, 1971).

Tanaka, C., L. Abel, and K. Naemura. "Parallel Logic Simulator and Its Use for Test Generation." ILLIAC IV Document No. 222, ILLIAC IV Project, University of Illinois at Urbana-Champaign (July 13, 1970).

6. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES

(Supported in part by the National Science Foundation under Grant No. US NSF GJ-812).

6.1 Computerized Mathematics

The research described in previous progress reports were collected into a Ph.D. thesis which was accepted by the Graduate College in June. The author received his Ph.D. on June 19, 1971.

We were able to show completeness of a very general form of the naming rule (see previous report). However, such a general form of the rule is not computationaly feasible. Thus, the existence of a complete and practicle form of the naming rule is still open. We were able to demonstrate a restricted form of the naming rule which was sufficient for an admitedly small number of sample problems.

(L. J. Henschen)

6.2 Computational Geometry

During this quarter, a DCL Report* was written and submitted for publication. A summary follows:

> An intersection detection procedure for polyhedral objects by means of face-to-face intersection analysis and its application for a path-finding problem in geometrically constrained space.

(K. Maruyama)

*Maruyama, K., "A Procedure for Detecting Intersections and Its Application," Department of Computer Science Rept. No. 449, 1971.

6.3 Educational Timesharing System on the PDP-11

ETS is designed to provide an interactive computing facility for non-sophisticated or beginning assembly language users. It will operate in a timesharing mode with 12-16 students who are connected to the processor via KSR-33 Teletypes. During its first semesters of operation it will support a teaching package called GIZMO (a question-answer routine used to introduce fundamentals to the student) and an on-line programming facility similar to DOS. Students will be able to write a program with an on-line editor, assemble it with a modified PAL-11R and run it under a powerful interpreter-debugging package. Concurrent with this operation, ETS will provide batch-type assemble and go facilities for students whose programs are on card decks.

The basic monitor is a revision of DEC's RSTS (formally BTSS) monitor (before the addition of the editor-compiler). The assembler is a modification of PAL-11R V000A. The editor, interpreter, and GIZMO are written locally. Our broad philosophy rests on making the system as simple as possible for the students while remaining compatible with DEC software specifications-conventions.

The monitor itself will be 6K words in size, leaving a 6K user area. However, it seems unreasonable to expect to operate with any efficiency where it is necessary to give the entire user area to a single task (PAL-11R will require 6K). A minimum job-mix seems to be of the order of 1 assembly and 2 edits (2K each) or equivalent (GIZMO = 2K, INTERPRETER = 2-3K). Thus a 10K user area would be needed. For this we need an additional 8K of core memory. (We presently have 8K and are in the process of adding an additional 4K).

As it is necessary to constantly maintain student interest, using the KSR-33 for long listings is not feasible. Output of PAL-11R must go to a line printer. Also, the line printer is to be the output device for batch operation. DEC has given us a line printer which will be installed as soon as possible.

The editor and GIZMO are both operational and are being refined for use this fall. Lessons for GIZMO are also being written. Plans call for 6-10 lessons of 60 questions to be written.

(D. B. Gillies)
(T. Chen)
(D. W. Oxley)
(A. Davis)
(S. Hansen)
(M. Stone)
(R. Atkinson)

6.4 Debugging Interpreter/Trace System (DITS)

DITS is being developed by the Computer Science Department of the University of Illinois as an educational tool for machine language programming for the PDP-11. It will be used interactively, primarily by students learning systems programming. By focusing on this class of user we hope to develop a very powerful debugging system, and gain insight into languages and debugging techniques in general.

Since student jobs are short, and are nearly all debugging runs, and since our PDP-11 is a fast dedicated resource it is feasible to run most, if not all such jobs interpretively. As well as interpreting every instruction and then executing it, DITS can perform certain tests, and maintain use tables and other statistics, automatically or as a result of user commands. The more it knows of the prcgrammers intentions, the more effectively it can monitor whether the program carries these out.

The goal is to present the student with a virtual machine which is instrumented, introspective, intelligent and interactive.

Many such systems are possible, and their design and testing is a promising research area. The choice of one to try first should be made within the general context of what is possible and desirable. Notice that syntax errors detected by an assembler or compiler resemble conventional computer-aided instruction since the computer can verify mechanically and directly from the form of the answer whether it follows the syntax rules. DITS is an example of a different and more demanding instructional use of the computer, namely the student designs a complex structure (in this case a program) whose correct functioning cannot be deduced from its form alone--it must be obeyed or simulated. This type of instrumented simulation requires a large number of instruction executions, each of a very simple type, and can be done more economically by a mini-computer or a network of mini-computers than by a large centralized computer.

Clearly, the more information available to DITS, the more sophisticated its checks can be. Such information can be passed through programming conventions, extensions to the assembler (both what it will accept and what it will tell DITS), user declarations and commands before and during interpretive execution, and user response to DITS typed enquiries. User commands in DITS correspond to a more or less conventional on-line debugging system except that more information is maintained by DITS and can be examined retrospectively by the programmer. Other DITS activities, using its knowledge derived from programming conventions, declarations, stylized comments, etc. are intended to flag suspicious or incorrect actions and outcomes automatically. The user is free to suppress or modify such testing or printing. Furthermore, the user has the option to run under DITS, partly under DITS or independent of DITS as he chooses.

6.5 PDP-11 Hardware Progress Report

Considerable progress has been made on PDP-11 hardware over the past quarter. An 8-bit extension to the processor status (PS) word and a clock were added to the PDP-11 processor. Detailed design of the relocation hardware was completed by Chuck Hyde and preliminary drawings were produced. The drawings were sent to C. Hyde for verification, were modified slightly, and have been returned. Priorities have pushed back the target date for working relocation hardware to Mid-September.

Production runs of the TTY interface boards were received from the printed circuit shop. A total of 54 boards were stuffed and soldered by the PDP-11 staff. Inventory and status of the TTY interface boards as of July 25 is shown below:

		ON HAND	Not Working
Transmitter/Receiver Boards	(Double Height)	20*	6
Interrupt Boards	(Single Height)	20*	2
Addressing Boards	(Single Height)	20*	3

* Includes 2 DEC boards

Five TTY's are currently on line and seven additional units have been allocated for PDP-11 use. These units will require some checkout and rewiring and will be installed when this work is completed.

The design of the interface for an Ampex 4K memory mod is complete and the boards are being wire-wrapped. The additional 4K should be operational by the 1st of August. This will bring total core up to 12K.

DEC has donated a Data Products Model 2410 line printer to the project. This is a 132-column drum type impact printer with a speed range of 245-1110 LPM. The printer should arrive about the 1st of August and will require about a week for checkout and installation. Prospects are grim for obtaining an additional 8K core, a card reader, and perhaps another disk platter before the fall term begins. Humidity and noise control in Room 123 are still significant problems.

(J. D. Miller)

6.6 Factorization Techniques Used in the Solution of Partial Differential Equations

We continued to work on a new algorithm which uses factorization methods to solve the large sets of algebraic equations that arise in the solution of partial differential equations by implicit numerical techniques.

Factorization methods solve the equation

$$AX = q$$

as follows.

An auxiliary matrix, B, is added to A so that (A+B) factors into known sparse triangular matrices L and U. The solution X is then given by the limit of the sequence $\{X_n\}$ generated from

$$(A+B)X_{n+1} = (A+B)X_n - \tau_n(AX_n - q).$$

The values of τ_n are determined from the values of the extreme eigenvalues of $(A+B)^{-1}A$.

The algorithm developed last quarter simultaneously solves $AX = q$ and generates an approximate eigenvector of $(A+B)^{-1}A$ corresponding to an extreme eigenvalue. This approximate eigenvector is used to calculate an approximate eigenvalue of $(A+B)^{-1}A$. The approximate eigenvalue is then used to improve the sequence of τ_n's.

Let a be an approximation to the smallest eigenvalue of $(A+B)^{-1}A$ and b be an approximation to the largest eigenvalue of $(A+B)^{-1}A$.

The approximate eigenvectors y_n satisfy

$$y_{n+1} = P_n((A+B)^{-1}A)y_1$$

where $$P_n(X) = T_n(\frac{a+b-2X}{b-a})/T_n(\frac{a+b}{b-a})$$ and $T_n(X)$

is the Tchebychev polynomial of degree n. The approximate eigenvalue is given by

$$U = \frac{< Ay_{n+1}, y_{n+1} >}{< (A+B)y_{n+1}, y_{n+1} >} .$$

It was shown that the convergence of y_n to an eigenvector corresponding to the largest eigenvalue is most rapid if a is equal to the smallest eigenvalue and b is equal to the second largest eigenvalue. It was also shown that the convergence of the eigenvalue was optimal if a and b are as just defined and

$$U = \frac{< Ay_{n+1}, y_{n+1} > - \frac{1}{T_n^2(y)} < Ay_1, y_1 >}{< (A+B)y_{n+1}, y_{n+1} > - \frac{1}{T_n^2(y)} < (A+B)y_1, y_1 >}$$

where $y = \frac{a + b}{b - a}$.

This result unfortunately has no practical application in the algorithm developed since it is assumed that nothing is known about the eigenvalue. Thus a and b cannot be chosen as above.

It was shown that for some special cases the elements of the auxiliary matrix defined by Stone's symmetric factorization [1] approach limits as the dimension of the matrix increases.

(M. A. Diamond)

[1] Stone, H. L., Private Communication, April, 1969.

7. THEORY OF DIGITAL COMPUTER ARITHMETIC

(Supported in part by the National Science Foundation under Grant No. US NSF GJ 813.)

7.1 Evaluation of Some Elementary Functions in Radix 16

During the quarter we have studied basic aspects of radix 16 iterative procedures, based on a continued product formulation, in order to determine the efficiency of this approach. Out interest has been restricted to four operations, namely division, multiplication, natural logarithm and exponential. It has been previously shown [1] that the computational procedures based on use of a continued product representation offer compatible algorithms with similar hardware requirements for elementary functions, making this computational approach very attractive in the view of new technological possibilities.

In a continued product form numbers are represented as $\prod_{k=0}^{m} (1 + r^{-k}s_k)$, where r is the radix and s_k is a properly chosen coefficient. In general, the hardware structure required for basic recursions in all algorithms under consideration, can be divided in three parts: adder complex, shifting network, and mechanism for selection of coefficients s_k. For radix 16 methods, the decrease of hardware requirements for the shifting network approximately equals the increase of adder complexity with respect to radix 2. Since the selection of s_k's is sufficiently simple after the first 3 interative steps, the increase in speed may justify implementation of radix 16 methods, provided that the selection of s_k's during the initial steps does not require a complicated procedure. It has

[1] B. DeLugish, "A Class of Algorithms for Automatic Evaluation of Certain Elementary Functions in a Binary Computer", University of Illinois, DCS Report No. 399, (Ph.D. Thesis), June 1, 1970.

been shown that starting selection rules can be implemented without any substantial increase in hardware. The requirement for ROM capacity for the set of precomputed constants which are needed in algorithms for natural logarithm and exponential, is increased 3 times compared to radix 2 methods. This is still acceptable the required size of ROM being approximately 2.5 (m + 5), words where m is the word length.

(M. D. Ercegovac)

7.2 Continued Fraction Arithmetic

During the quarter effort was directed toward the solution of the set of quadratics

$$x^2 + b_k x - c_k = (x-u)(x-v) = 0$$

such that $\frac{1}{2} \leq u \leq 1$. The problem, specifically is, given b_k and c_k, find u. The method consists of expanding u in the form of a periodic continued fracti and gradually increasing the period of the fraction.

In an earlier report[*], v was restricted to a small range and the square rooting problem was only soluble for $c_k = 0$. On reflection it was seen that such a restriction on v is not necessary and hence a subset of the set of quadratics, namely all those with $b_k = 0$, (that is, the square rooting problem) was made the target of investigation.

With the given limits on u the triangular region enclosed by $c_k - b_k = 1$ and $c_k - \frac{1}{2} b_k = \frac{1}{4}$ was the region of the c_k, b_k plane that is amenable to solution by this method.

[*] Quarterly Progress Report, Department of Computer Science, University of Illinois, June-September, 1970, pp. 132-137.

The recursion is given by

$$b_{k-1} = q_1 c_k$$

$$c_{k-1} = 1 - q_1(b_{k-1} - b_k)$$

$$b_{k-n} = q_n\ c_{k-n+1} - q_{n-1}\ c_{k-n+2} + b_{k-n+2}$$

$$c_{k-n} = -q_n\ b_{k-n} + q_n b_{k-n+1} + c_{k-n+2}$$

$$P_n = q_n\ P_{n-1} + P_{n-2}$$

$$Q_n = q_n\ Q_{n-1} + Q_{n-2}.$$

The problem then is the selection rules for partial denominators q_i.

With $q_i \in \{\frac{1}{2}, 1\}$, there was no freedom of choice of q_1 and the dividing line had an intercept of 2/3 (infinite binary fraction) on the $b_k = 0$ axis and hence redundancy in the digit set was required.

A choice of $q_i \in \{\frac{1}{4}, \frac{1}{2}, 1\}$ was made. The region of c_k, b_k plane soluble is now enclosed by $c_k - 1.56\ b_k = (1.56)^2$ and $c_k - 0.39\ b_k = (0.39)^2$. In particular this means that the square rooting problem can be solved for $0.15 < c_k < 2.4$.

A set of selection rules for the choice of partial denominators has been developed, with extensive numerical simulations indicating their validity. A complete proof of convergence of the algorithm with these selection rules is the future aim of the investigation.

(Kishor Trivedi)

8. SWITCHING THEORY AND LOGICAL DESIGN

(Supported in part by the National Science Foundation under Grant Number U.S. NSF-GJ-503.)

When the branch-and bound method is applied to logical design, the method generates feasible solutions successively (the last one is an optimum network). We have been trying to improve the efficiency of the method, by examining whether each feasible solution contains redundant gates or interconnections. If they do, we can skip intermediate feasible solutions, shortening computation time. Preparation of a computer program to draw networks of feasible solutions facilitates discovery of procedures for this redundancy check.

Integrated circuitry provides logical designers with extra means such as wired logic. We started to look at the logical design with wired logic capability.

Ph.D. thesis research by T. K. Liu has been in good progress. Logical design with MOS is explored.

S. MUROGA

Our program for drawing networks of gates by Calcomp was extended to include NAND, EXCLUSIVE-OR, and EXCLUSIVE-NOR gates. Also, the number of allowable external variables was increased from 5 to 9. A user's manual has been prepared to allow the use of the program by other researchers.

Network transformations were studied with the idea of transforming (within a reasonable time) any network which realizes a given function into an optimal or near optimal network which realizes the same function.

Only minor results have been obtained thus far.

(J. Culliney)

There are several logic families of integrated circuits which have capabilities of wired logic. A NOR gate with wired OR capability (or a NAND gate with wired AND capability) is widely used because it is realized by DTL, some kind of TTL and ECL. By utilizing the wired logic capability, the number of gates required to realize a given logic function can be reduced. Properties of a network using this kind of logic gates were obtained. By integer programming we could obtain all minimal networks of three-variable functions using NOR gates with wired OR capability (all minimal networks using NAND gates with wired AND capability are obtained by a simple transformation). This kind of minimal networks were not known before.

(Y. Kambayaski)

An interface program between the program ILLOD(NOR-B) and Jay Culliney's CALCOMP program (see Quarterly Report for Jan., Feb., March, 1971 by J. Culliney) was developed and debugged. Using the solutions on magnetic tape obtained by ILLOD(NOR-B) as the input data, this program can draw the corresponding network of each solution. The user of this program can specify the problem numbers or the solution numbers under a problem number. In the former case, all solutions of those problems will be drawn; and in the latter case only the specified solutions of that problem will be drawn.

(H.C. Lai)

How to realize a negative function by a single MOS cell which consists of the minimum number of FET's has been investigated. This problem is similar to the classical relay network synthesis problem. However more restrictions can be added due to some properties of the MOS cells. If bridge connections are prohibited in the cells, this problem can be formulated into a tree type AND/OR gate network, and solved by the branch-and-bound method. The structure of optimum cells is being studied. It is hoped that some properties of the optimum cells can be incorporated with the branch-and-bound method in order to speed up its convergence.

(T.K. Liu)

Publications during this quarter

Nakagawa, T., and H.C. Lai, "A Branch-And-Bound Algorithm for Optimal NOR Networks (The Algorithm Description).

Nakagawa, T., and S. Muroga, "Comparison of the Implicit Enumeration Method and the Branch-And-Bound Method for Logical Design," Department of Computer Science, University of Illinois, Report No. 455, June, 1971, 94 pages.

Nakagawa, T., "A Branch-And-Bound Algorithm for Optimal AND-OR Networks (The Algorithm Description)," Department of Computer Science, University of Illinois, Report No. 462, June, 1971, 39 pages.

9. MACHINE AND SOFTWARE ORGANIZATION STUDIES

The following is a collection of related work aimed at improved designs for computer and software systems. We are interested in parallel processors, small primary memories, effective use of rotating memories and some questions concerning user languages for problems including typical Fortran type calculations and certain file processing problems.

9.1 Fortran Parallelism Detection (C. Cartegini, S. Chen, J. Han)

The function of the scanner has been refined. One conflict was the handling of variable names inside a DO-loop used simultaneously as a fixed subscript in an array-name. A similar conflict was the occurrence of the DO-loop index as a single variable name. Identical local names in decimal form have been provided. The scanner can now support more general Fortran programs with the exception of continuation cards.

For the purposes of a refined and more general analysis of Fortran programs the detection of IF-trees has become an important feature. The purpose of IF-tree detection is to handle them in a different way so that their occurrence does not mean an overall loss of efficiency. It may be recalled that the definition of an IF-tree supposes a high ratio of IF (logical, arithmetical) and GO TO statements to assignment statements. The new program which is being written should allow a more effective handling of the data so that retrieval will not cause the searching of large tables. The structure used for storing the data pertaining to each single block is a list-structure. The features for passing the structure elements from one procedure to another are being tested.

Some refinements of the DO Loop Subprogram have been made during this quarter. This includes a modification of the basic program and data structure in order to be able to handle more cases in general DO loops, such as the index variable used in an arithmetic expression within the same loop and any non-index variable used as a subscript. This reprogramming structure could lead to the possibility of extracting more parallelism within DO loops by backsubstituting the non-index subscript with its original form. For example, by transforming <DO I=1,10; N=I+1; A(I)=A(N)+K > to the form <DO I=1,10; N=I+1; A(I)=A(I+1)+K >, the parallelism which exists in the statement A(I)=A(I+1)+K can be exploited explicitly by the detecting algorithm employed in the subprogram. For this purpose, a subroutine which can do this simple first-level backsubstitution has been implemented and tested, and will be integrated into the main subprogram.

Also, some preliminary experiments on real application programs have been done by combining the DO Loop Subprogram with the Masterprogram and Assignment Statement Subprogram. During the experiment, some unexpected cases raised in a real program were found and debugged. A larger group of application programs taken from the University of Illinois subroutine library will be under experiment next.

For the complete analysis of DO loops, a further step is being taken to consider IF statements within DO loops. Some strategies have been considered. Since the detecting logic used in the reprogrammed subprogram could be compatible to this goal, the results are still expected to be feasible.

For assignment statements the implementation of the tree height reduction algorithm has been programmed in PL/1 and tested (see the last quarterly report). In order to cooperate with the masterprogram and DO-loop subprogram to measure Fortran programs for parallel processing, some modifications are desirable such as omitting the building of syntatic trees for assignment statements to obtain less program execution time. Now a new PL/1 program has been programmed and tested. This program consists of manipulations of strings of numbers and related numbers. Those are minimum tree height, effective length, multiplicative length and absolute holes associated with each term in the assignment statement. They are required to compute the minimum tree height and minimum number of PE's required to execute the assignment statement within the time of the tree height. In addition to this, the number of operations for each level are counted in such a way that Hu's algorithm can be applied to obtain the minimum number of PE's for a tree graph within the time limit of the tree height.

For blocks of assignment statements the tree height reduction algorithm is being expanded to accommodate the back substitution and recursion algorithms properly to obtain a lower tree height and more parallelism.

By the application of back substitution the sequential relations among assignment statements may be changed into a parallel relation, i.e. they may be independent of one another and can be executed simultaneously in parallel machines.

By the application of recursion, iterative assignment statements in DO-loops might be handled as a set of independent assignment statements, rather than the iterations of sequential operations. Sometimes in this way we might gain the advantage for parallel operations.

9.2 Weighted Node, Directed, Acyclic Graph Schedule by m-Machines (P. Kraska)

We have developed a scheduling algorithm for m machines on a weighted node, directed, acyclic graph which is probably optimal, or near optimal, this quarter; the analysis is not yet complete. However, the following lemma was found which provides a lower bound on the number of machines required to process a graph in the critical time, W_q, of a graph G:

Let G_R be the relaxed graph of G (i.e., settle each node so that it is tightly connected to the terminal node, n_t) and label each level of independent nodes 1, 2, ..., q where n_t is at level 1. Let P(j) be the sum of the node weights (of independent nodes) at level j and W_j be the maximum longest path (i.e., the critical path) of node-weight sums of all nodes in level q+1-j to the starting nodes. W_q is the critical path time of G_R (and G) to the terminal node, n_t.

<u>Lemma</u>: If $n < \max\limits_{k} \dfrac{\sum\limits_{i=1}^{k} P(q+1-i)}{W_i}$ then it is impossible to process all nodes with n machines in W_q time.

We now sketch the proposed scheduling algorithm. Suppose we have found m, the smallest number of machines required to process the graph in W_q time ($m > n$). For each node $n_i \epsilon G$ calculate W_{it}, the largest node-sum on all paths from n_i to n_t the terminal node. Of the initial set of starting nodes assign up to m with the largest W_{it} to the machines; as time advances and k machines become free ($k \leq m$), update the set of starting nodes and again assign up to k of these with the largest W_{it} to the machines. The rationale for the algorithm's optimality is that the time-dependent critical path is guaranteed to emanate from among the nodes being processed at all times, for a sufficiently large m.

Methods of factoring common subexpressions from tree-height reduced parses of arithmetic expressions, thereby transforming the parse-tree to a graph, were also investigated. An interesting fact was uncovered about the number of operations required when we factor common subexpressions from polynomials which have been parsed in accordance with Muraoka's folding method [1] which produces a tree-height near the limit of $1+\log_2[n]$. That is, we write

$$P_n(x) = a_0+a_1x+a_2x^2+(a_3+a_4x)x^3+(a_5+a_6x+a_7x^2)x^5+\dots$$

and we find that the number of operations required, y, is bounded above, i.e.,

$$y < 2n+\log_f(n)+1 ,$$

where f is the Fibonacci ratio 1.618... We may write this more conveniently as $y < 2n+4.79\log(n)+1$. This means, for example, that if $n=10^6$ then less than 30 operations are required in excess of 2n, the minimum as prescribed by Horner's rule. (The folding method tree-height for $n=10^6$ is 30 compared with 21, the minimum.)

9.3 Simulation Processor (E. W. Davis)

The control unit, which resolves run time precedence questions, has been designed at the register level. A simulation of the control unit has been written in GPSS. This simulator will be used to study the performance of the unit on simulation programs being gathered from industry and university sources.

Organization of the processing units is being studied. Simulation programs have been found to consist of simple assignment statements intermixed with decision statements such that the ratio of IF's to assignments is approximately one. For this reason the design of the processors will include special logic for program control.

9.4 Memory Hierarchies (D. Gold)

Recent results indicate that any array operation may have its data mapped on the disk in such a way to yield no disk latency when accessing it. This is true even if only a subset of the original data is used and if the data is accessed differently on successive iterations.

The significance of the above is that for the array operations of Fortran-like programs, zero latency data mappings can be obtained for a small primary memory system. It is, however, still necessary to obtain a means of abstracting the relevant array index information from the original (source code) Fortran program.

9.5 Microprogramming and Control Unit Design (L. A. Hollaar)

The production version of the Burroughs microprogrammed machine was delivered during June. Work prior to this time consisted of preparing for its arrival. The indicator and switch panels were removed from prototype 3 and mounted in the new machine's rack; by switching the cables plugged into them, either machine can be used.

The individual boards of the new machine were tested by Burroughs, but the interconnections on the backplane were not. At the present time, the only wiring errors encountered were caused by human error or incorrect input to the automatic wire-wrap program; no mistakes by the wiring machine have been found.

The various input/output buses and control lines have been added to the backplane of the machine. Currently, the machine can be connected to four perpherial devices at one time.

Work has started on the connection of the microstore, which consists of four 1024 by 16 bit Cogar memories, to the control lines of the machine. The cycle time of the memory and the machine will be 400 nanoseconds (2.5 MHz).

After the microstore has been installed and tested, the various sections of the machine will be debugged, a core memory will be added, and the card reader and printer (currently in use by the PDP-11 project) will be interfaced. Following that, work on an automatic microstore loader, the disk interface, and a broadbank link to the IBM System/360 is planned.

9.6 S-Language Assembler (E. Polley)

The string instructions for the assembler are in the process of being implemented into the language. Meanwhile test programs consisting of only word instructions are being run to locate any bugs in the code for those instructions.

In the final form the assembler will punch object decks (on the 360) which will be loaded into the D-machine for interpreting. Object code is currently being printed for each instruction but is not as yet punched. As soon as some longer programs are written in the S-Language timing estimates will be available on the speed of the assembler.

9.7 Text Searching (W. Stellhorn)

Implementation has begun on a preliminary version of the interactive language for text searching. The initial program, to be used for experiments with search and data organization techniques, timing requirements and user reactions, will contain most of the features desired in later versions for controlling the search process and displaying the results.

The user will be allowed to request a search for any desired character string(s), and he may control the context of the search (full text, title, author, etc.). He may also specify that two or more strings should occur together within a given context (sentence, paragraph, article, title, etc.) or that documents which contain any of several strings or groups of strings are to be retrieved. Similar flexibility will be provided for controlling the output from a search. The user may request printing of any or all of the following: full text, selected bibliographic information, or the sentence or paragraph in which a "match" was found.

Background investigations are continuing in a number of pertinent areas including automated information retrieval, interactive languages and the production of digitized text.

Several items of B5500 software support for the D-machine have been received from the Burroughs Corporation and have been partially tested on local facilities and made available for use. These include a simulator for assistance in developing and testing microprograms, and several versions of the microcode assembler.

9.8 Debugging (M. Kaplan)

Implementation of the debugging package described in the last Quarterly Technical Report continued with the target date for implementation on the University of Illinois EXPRESS system as the beginning of the fall semester.

Version I Release 2 of WATFIV, which went on both the HASP and EXPRESS systems in June contains some new debugging features provided by Waterloo. These include a flow-of-control trace, the ON ERROR GO TO statement and DUMPLIST. (Details may be obtained from the consultants.) The first two of these are similar to statements in the proposed debugging package (LINETRACE and ON ERROR) and make the implementation of these unnecessary.

REFERENCES

[1] Muraoka, Y., "Parallelism Exposure and Exploitation in Programs," Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 424 (February, 1971).

10. COMPUTER SYSTEMS ANALYSIS

(Supported in part by the National Science Foundation under Grant No. US NSF GJ 28289.)

The goal of this research is the development of analytical tools for system modeling and analysis of real time computer networks. The particular network being investigated is that of a geographically distributed network of computers. A queueing theory model for this computing system based on the essential characteristics of the network, and priority assignment rules for efficient job processing at each of the computing centers of the network are being investigated.

10.1 Computer Network Modeling (W. Barr)

Our current effort in this area is aimed at developing a queueing theory model for a multiserver system with a finite length queue. To date we have considered nonpriority and nonpreemptive priority queue disciplines. We are currently investigating preemptive priority and dynamic priority queues.

Additionally, we are working to modify existing mathematical models of communication nets so that they more accurately represent our network. Cost functions for communication within the net are being developed. Once our model is completely defined we will look at criteria for inter-center load transmitting for optimization of throughput for the entire network. We also plan to investigate techniques for minimizing congestion within the network.

10.2 Center Throughput Analysis (J. Fitzgerald)

Our research in this area is aimed at analyzing Illinet, a geographically distributed computing center which provides online express, teletype timesharing, and remote batch entry services to a network of users at the University of Illinois. We are currently determining the essential characteristics of the various computers within the center.

We plan to employ GPSS (General Purpose Simulation System/360) in conjunction with our queueing theory model to facilitate studying the effects of priority assignment, job dispatching, and load regulation within the network.

(E. K. Bowdon)

11. SOUPAC AND STATISTICS

(Statistically Oriented Users Programming and Consulting)

April - June 1971

11.1 SOUPAC Consultants

A version of Soupac called Soupex is available on Express. This statistical system has the capability for:

- Means and Standard Deviations
- Pearson product moment correlations with optional missing data facility
- Principal axis factors and Varimax rotation
- One-way frequency counting

Soupex is limited to 30 variables and all the other limiting requirements of the Express System. It was felt users would appreciate the instant turnaround and simplified program cards of such a system.

A further use of express to codecheck a standard Soupac program deck is ready and will soon be available to users. The facility called Soupscan will check for keypunch and other errors and print diagnostics. Users may then correct their programs and run them with data (if any) on Hasp. Soupscan is a codechecking procedure and does not execute a Soupac program.

Soupac has been implemented at Cleveland State University in Ohio, at the National Headquarters of Girl Scouts and Megasystems Corporation in New York City and at the Bureau of Labor Statistics in Washington, D.C..

The Soupac Staff began maintaining a Quadratic Programming program from the Rand Corporation made available through the Agricultural Economics Department. A series of Multidimensional Scaling programs from the Psychology Department is also available through the Soupac Consultants.

During the period (April to June 1971) 13, 451 jobs were run in Soupac. This is an average of over 4,460 jobs per month or nearly 150 jobs per day.

11.2 Research

Several goodness of fit measures for factor analysis were studied and applied to the results of maximum-likelihood factor analysis. In order to do this, 96 sample correlation matrices were drawn, twelve from each of eight population-correlation matrices with known factors. Each sample correlation matrix was factored by the maximum-likelihood technique, and the results were tested for goodness of fit. Each of the 96 matrices had 20 variables, and they were arranged in a 2 x 2 x 2 x 3 ANOVA design with three observations per cell. The four independent variables were number of factors (3 or 7), level of communality (high or low), model (whether or not the factor analytic model held in the population), and sample size (100, 400, and 1600).

Results showed that ρ_2 was a fairly good measure of the goodness of fit of the factor analytic model in the population. Although the sample values were somewhat large for small sample sizes, they approached the population values as the sample size increased. Two other measures, c_1 and r_1, were used to compare the obtained sample factors with the population factors. These two measures can be used whenever a hypothesis factor matrix can be written down. They showed that fairly good agreement with an hypothesis can be obtained, even though the factor analytic model does not exactly hold in the population. This would require that the experimenter have variables with high communality and have a low ratio of factors to variables.

In general, better fit was obtained when the factor analytic model held exactly in the population, when there were three rather than seven factors, when the variables had high communality, and with larger sample sizes.

The above work was reported in DCS Report No. 451 An Investigation of the Goodness of Fit of the Maximum Likelihood Estimation Procedure in Factor Analysis. Some of the results are now being used in a comparison of several factor analytic techniques.

12. GENERAL DEPARTMENT INFORMATION

12.1 Personnel

The number of people associated with the Department in various capacities is given in the following table:

	Full-time	Part-time	Full-time Equivalent
Faculty	20	3	21.77
Visiting Faculty	4	0	4.00
Graduate Research Assistants	2	55	29.33
Graduate Teaching Assistants	0	28	13.11
Professional Personnel	9	1	9.50
Administrative and Clerical	19	0	19.00
Nonacademic Personnel (Monthly)	17	1	17.50
Nonacademic Personnel (Weekly)	0	32	18.91
TOTAL	71	120	133.12

*This report does not include personnel employed by the Computing Services Office.

The Department Advisory Committee consists of Professor J. N. Snyder, Head of the Department, Professors E. K. Bowdon, D. F. Cudia, K. W. Dickman, H. G. Friedman, C. W. Gear, D. B. Gillies, D. J. Kuck, B. H. McCormick, S. Muroga, T. A. Murrell, J. Nievergelt, J. R. Phillips, W. J. Poppelbaum, S. R. Ray, E. M. Reingold, J. E. Robertson, P. E. Saylor, D. L. Slotnick, and D. S. Watanabe.

12.2 Bibliography

During the second quarter, the following publications were issued by the Laboratory:

File Numbers

(1) Borovec, Richard T., "IMAGE 8: The Real Time Clock," File No. 863, June 16, 1971

Report Numbers

(1) ILLIAC IV Quarterly Progress Report, January, February, and March 1971, Report No. 456, April 15, 1971

(2) Maruyama, Kiyoshi, "A Procedure for Detecting Intersections and Its Application," Report No. 449, May 1971

(3) Michalski, R. S., "A Geometrical Model for the Synthesis of Interval Covers," Report No. 461. June 24, 1971

(4) Michalski, R. S., and B. H. McCormick, "Interval Generalization of Switching Theory," Report No. 442, May 3, 1971

(5) Nakagawa, Tomoyasu, and Hung-Chi Lai, "A Branch-and-Bound Algorithm for Optimal Nor Networks," (The Algorithm Description), Report No. 438, April 1971

(6) Nakagawa, Tomoyasu, and Saburo Muroga, "Comparison of the Implicit Enumeration Method and the Branch-and-Bound Method for Logical Design," Report No. 455, June 1971

(7) Ratliff, K., "Sparse Matrix Inversion," Report No. 443, June 1971

(8) Reingold, Edward M., "Notes on AVL Trees," Report No. 441, May 10, 1971

(9) Reingold, Edward M.. "A Destructive List Copying Algorithm," Report No. 454, June 1971

Theses

(1) Alsberg, Peter Allyn, "OSL/2: An Operating System Language," (Ph.D.), Report No. 460, June 10, 1971

(2) Beals, Alan James, "The Automatic Generation of Deterministic Parsing Algorithms," (Ph.D.), Report No. 458, June 21, 1971

(3) Bracha, Amnon, "A Symmetric Factorization Procedure for the Solution of Elliptic Boundary Value Problems," (Ph.D.) Report No. 440, April 30, 1971

(4) Garton, Richard Dean, "A Hardware Realization of a Decomposition Algorithm," (M.S.), Report No. 457, June 1971

(5) Guimaraes, Celso John Frazao, "WEED: A Wonderful Equation Elimination Device," (M.S.), Report No. 444, June 1971

(6) Henschen, Lawrence Joseph, "A Resolution Style Proof Procedure for Higher-Order Logic," (Ph.D.), Report No. 452, June 1971

(7) Jacobson, Robert Clarence. "Path Integral Calculation of the Three-Particle Direct and Exchange Contributions to the Pair Distribution Function and of the Third Virial Coefficient for HE^4 Gas," (Ph.D.), Report No. 445, June 1971

(8) LaFrance, Jacques Emmett, "Syntax-Directed Error Recovery for Compilers," (Ph.D.), Report No. 459, June 21, 1971

(9) Mark, Barbara Drahos, "Machine Independent Compilation of PL/I: Pass I Symbol Manipulation," (M.S.), Report No. 439, June 1971

(10) Montanelli, Richard G., Jr., "In Investigation of the Goodness of Fit of the Maximum Likelihood Estimation Procedure in Factor Analysis," (Ph.D.), Report No. 451, June 1971

(11) Rey, Christian A., "Control Point Strategy and Its Automated Diagnosis," (M.S.), Report No. 450, June 1971

(12) Ryan, Lawrence D., "System and Circuit Design of the Transformatirx Coefficient Processor and Output Data Channel," (Ph.D.), Report No. 435, June 1971

(13) Simons, Arthur, "Eidolyzer: A Hardware Realization of Context-Guided Picture Interpretation," (Ph.D.), Report No. 448, June 1971

(14) Wang, Paul Jang-Ching, "Machine-Independent Compilation of PL/I: Compilation," (M.S.), Report No. 453, June 1971

(15) Yamada, Hirohide, "Emulation of Disc File Processor," (M.S.), Report No. 436, June 1971

12.3 Colloquia

"Probabilistic Generative Grammars," by Mr. Stephen Soule, Committee on Information Sciences, The University of Chicago, April 1, 1971.

"Formal Models for Memory Management Problems," by Professor Peter J. Denning, Department of Electrical Engineering, Princeton University, April 12, 1971.

"ATC (Air Travel Control) Terminal Automation," by Mr. Francis S. Carr, Department of Transportation, Federal Aviation Administration, Washington, D.C. . April 19, 1971.

"The Posterior Intrinsic Systems of the Brain as a Jet Bundle," by Professor William C. Hoffman, Department of Mathematics, Oakland University, Rochester, Michigan, April 26, 1971.

"Lunar Surface Holography," by Dr. Donald H. Close, Hughes Research Laboratories, Malibu, California, May 3, 1971.

"Switching and Memory Effects in Amorphous Semiconductors," by Professor Hellmut Fritzsche, The James Franck Institute, The University of Chicago, May 10, 1971.

"Flow Analysis and Register Allocation," by Mr. Kenneth Kennedy, New York University, May 13, 1971.

"Information Work and Information Entropy," by Dr. R. R. Johnson, Vice President, Engineering, Burroughs Corporation, May 17, 1971.

"On the Algebraic Complexity of Matrix Multiplication," by Mr. Charles M. Fiduccia, Brown University, May 24, 1971.

"Models of Computation--Mathematical vs. 'Realistic' Properties," by Professor Edward Robertson, The University of Wisconsin, Computer Science Department, June 3, 1971.

12.4 Drafting

During the second quarter, a total of 465 were processed by the general departmental drafting section:

Large Drawings	122
Medium Drawings	91
Small Drawings	151
Layouts	3
Report Drawings	70
Changes	26
Miscellaneous	2
Total	465

(M. Goebel)

12.5 Shops' Production

Job orders processed and completed during the second quarter of 1971 are as follows:

	AEC 2118	AEC 1469	Other
Machine Shop	3	16	2
Electronic Shop	3	109	16
Chemical Shop	3	89	10
Layout Shop	0	72	8

(F. P. Serio)

~~Physics~~

COO-1469-0196
COO-2118-0025

QUARTERLY TECHNICAL PROGRESS REPORT

July, August, September 1971

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

QUARTERLY TECHNICAL PROGRESS REPORT

July, August, September 1971

UIUCDCS-71-3

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

TABLE OF CONTENTS

Page

O. DISTRIBUTION OF DEPARTMENT OF COMPUTER SCIENCE REPORTS

In order to achieve greater efficiency and more effective dissemination of information, some revisions in the method of distributing reports from the Department of Computer Science are being put into effect with the opening of the 1972 fiscal year.

The Quarterly Technical Progress Report will be the primary medium by which information about the department is disseminated. Normally, a person requesting the reports of the department will be placed on the mailing list for this report. Individual mailing lists for all the reports generated in the department or in any group thereof will be held to a minimum. However, each report generated will be listed at the end of the section pertaining to each group. A combined list of all of these reports will appear toward the end of the Quarterly Technical Progress Report. An addressed request sheet is also provided at the end of this report by which a selected number of departmental reports of interest may be requested.

The series of file numbers generated by the department normally contains material useful internally. In the rare event that one of these might be useful outside of the department, a request should be submitted directly to the person concerned.

Approximately once per year an inquiry form will be provided in the Quarterly Technical Progress Report via which the department can ascertain whether or not the recipients of the report wish to remain on the mailing list.

1. CIRCUIT RESEARCH

(Supported in part by the Office of Naval Research under Contract N000 14-67-A-0305-0007, W. J. Poppelbaum, Principal Investigator.)

Summary

The bundle repeater/restorer for the system that has been described by Bernard Tse is now being built, and the system should be complete shortly. SABUMA (Safe Bundle Machine) is also nearly finished; its properties are summarized by Trevor Mudge. Yiu Wo's report on APE (Antonomous Processing Element) deals mainly with the control unit and, in particular, with a control code generator for transmitting setup information to the assembly of elements. Panigrahi describes the transmit/receive sections of PENTECOST, the color display that uses the two color Land scheme, a "filtered" black and white TV camera and a Penetron display tube. In addition, two new projects are outlined in this section. Ergodic, tackled by Jim Cutler, aims to build a stochastic processor which combines both spatial and temporal aspects of random sequence coding to add redundancy to a system. The second of these is Telemaze, which uses a "world model" approach to get around unacceptably long feedback delays in remote space control systems. Basic to this is the idea that an earthbound model of the environment of a remote object (planetary lander) need be updated relatively infrequently by the controller, thus eliminating the need for the object to send back environmental information at all times. Edward Pott is the mission controller.

M. Faiman (ed.)

1.1 Bundle Processing (Project No. 21)

1.1.1 Bundle Repeater and Restorer

Construction of the bundle repeater/restorer is underway. Supplies for +5V and -5V had been acquired. The logic circuit for the repeater/restorer stages will be laid out by the shop since 384 similar circuits will be required.

It is expected that the repeater/restorer system will be completed during the present quarter.

Bernard Tse

1.1.2 SABUMA

Summary of Project:

This system demonstrates the feasibility of failsafe information interchange and processing using a bundle representation. The technique may be summarized as follows. Numbers in the range $-50 \leq n \leq 50$ are represented by the sum of the potientials on a bundle of 100 wires. A wire is weighted -1, 0, or +1 according to whether it is at a negative potential, open circuit (broken), or at a positive potential. Thus, $n = 32$ would be represented by having 32 more wires at a positive potential than at a negative one.

The information interchange is made failsafe by using these bundles in pairs, to represent numbers by the quotient of the two numbers represented by the two bundles. The new number generated is thus in the range $-1 \leq N \leq 1$. If one bundle represents X and the other, Y, then

$$N = \frac{X}{Y}$$

If the probability of a wire going open circuit, i.e. breaking, is P, then by the law of large numbers we have on average

$$N^1 = \frac{P.X}{P.Y} = N$$

i.e. failsafe action.

Furthermore assume we wish to perform arithmetic with two numbers represented in this bundle technique, say A = $\frac{X_1}{Y_1}$; B = $\frac{X_2}{Y_2}$; in general we need

(i) $A + B$

(ii) $A - B$

(iii) $A * B$

(iv) A/B

These are formed as follows:

(i) $$\frac{X_1Y_2 + Y_1X_2}{Y_1Y_2}$$

(ii) $$\frac{X_1Y_2 - Y_1X_2}{Y_1Y_2}$$

(iii) $$\frac{X_1X_2}{Y_1Y_2}$$

(iv) $$\frac{X_1Y_2}{Y_1X_2}$$

Addition between two bundles is trivial - form the new sum bundle from half of each of the addend & augend bundles. Subtraction requires the value of a bundle to be complemented i.e. positive potential must be made negative and vice versa. This is performed electronically. Finally multiplication must be achieved; this is done using a 3-valued logic gate having the truth table shown.

	+	0	-
+	+	0	-
0	0	0	0
-	-	0	+

TRUTH TABLE

FOR MULTIPLIER.

Project Status:

Construction is nearly complete. The remaining task is to install the displays. Subsequently to this the hardware must be debugged.

Trevor Mudge

1.2 APE (Project No. 25)

1.2.1 Progress in General

During the past quarter, much effort was put into the design and construction of the control unit. Design work on most of the circuits has been completed. Orders on most of the required active circuit elements, components, power supplies, cabinet and mounting hardware have been made. Printed circuit cards for the control unit are now being produced. In the following section a control code generator of the control unit is described in detail.

1.2.2. The 32-Bit Control Code Generator

An important function of the control unit is to send out information to each APE as to what operation it is to perform on its input data, as well as from what channels its inputs are tuned to acquire data. This information is coded into a binary pulse width modulated signal which is later used to modulate an RF signal for transmitting through a whip antenna. The function of the 32-bit control code generator is to produce these pulse width modulated control signals. The block diagram of this generator is given in Figure 1. It consists of a pulse sequence generator, a 5-bit counter, a 32-bit multiplexer, and a monostable multivibrator. The pulse sequence generator produces a sequence of 32 pulses when a starting signal is received. The 5-bit counter and the 32-bit multiplexer are connected to form a parallel to serial converter. The 32-bit control words are fed to the data inputs of the multiplexer. Then the 32 inputs of the multiplexer are gated sequentially to the output. The pulse sequence from the output of the multiplexer is used to modulate the timing network of the monostable multivibrator by switching an auxiliary timing resistor in and out of the timing network. The output

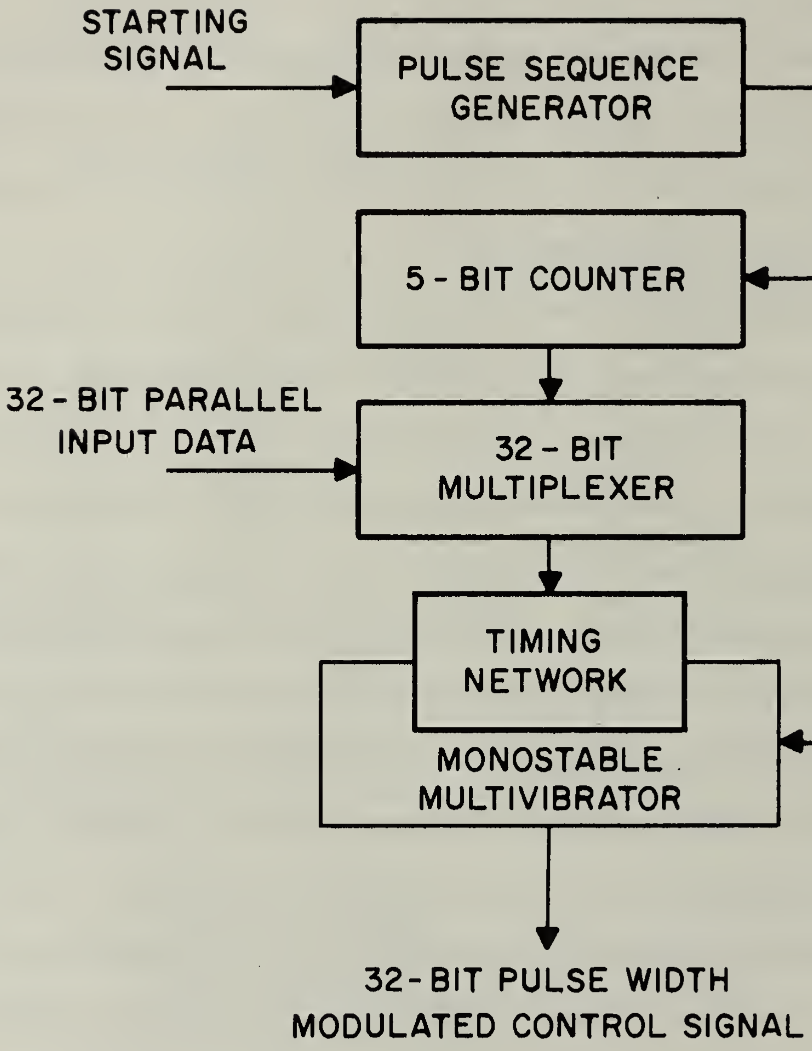
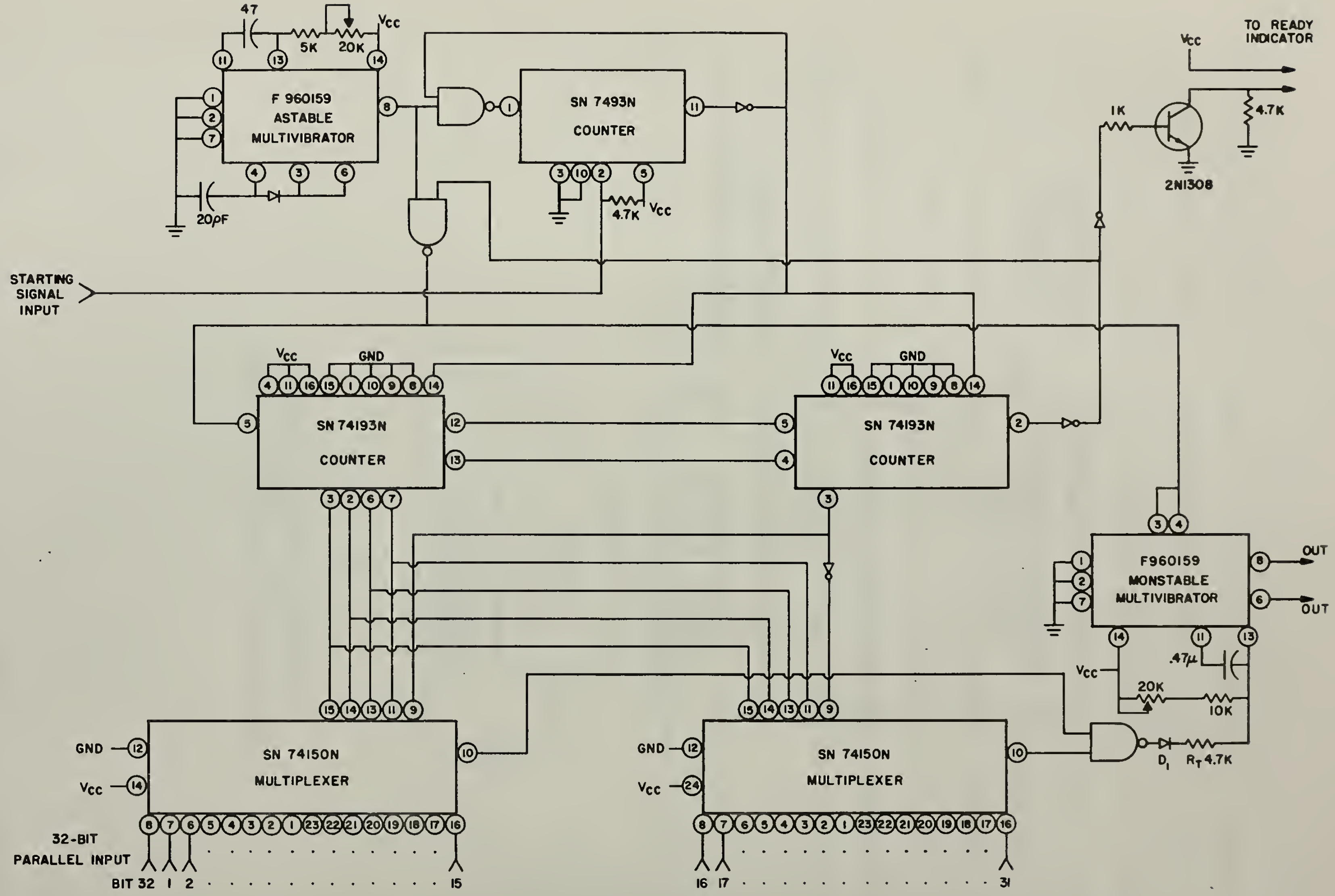


Figure 1. Block Diagram of the 32-bit Control Code Generator.

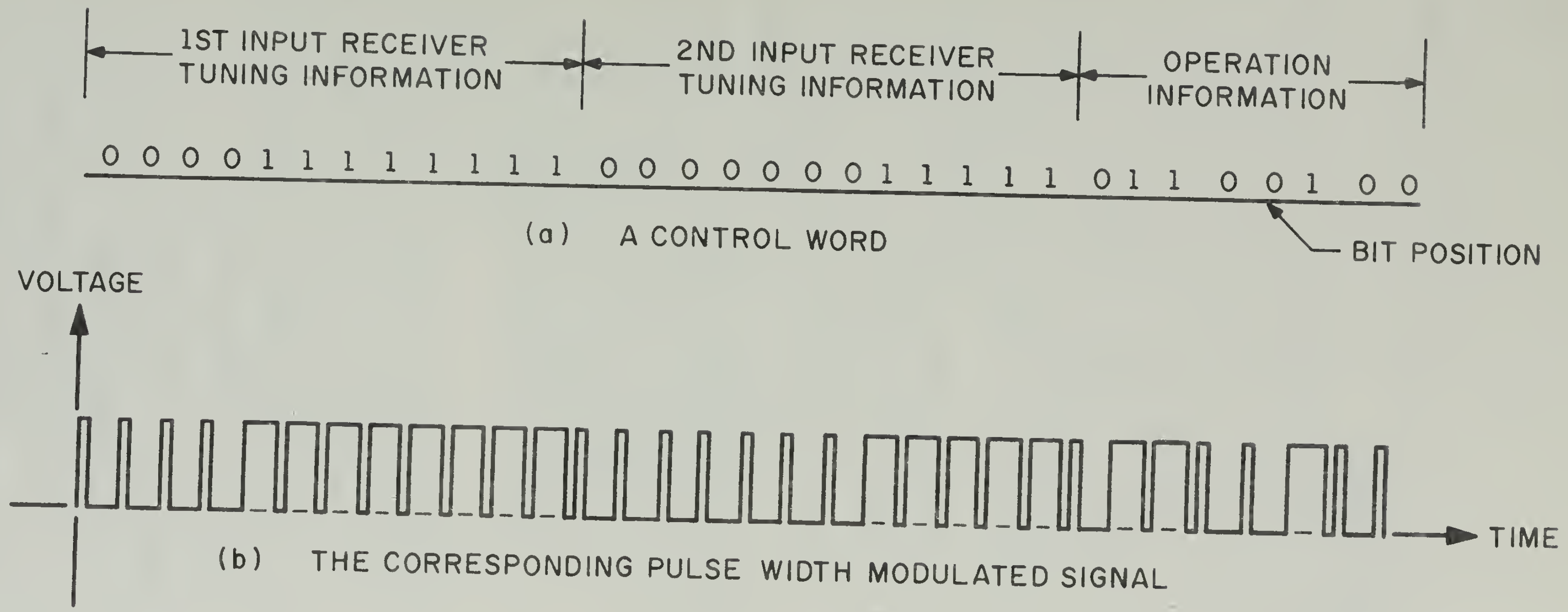
from the monostable multivibrator produces, therefore, a binary pulse width modulated signal. Figure 2 shows the actual circuit of the 32-bit control code generator. Two SN74150Ns are employed to form the 32-bit multiplexer. The diode D_1 is used to switch the 4.7kΩ auxiliary timing resistor R_T in and out of the timing network. The ready indicator circuit is to turn on an indicator light when the 32-bit control code generator has completed sending out a full control word and is ready for loading with another word for another APE channel. Figure 3 shows a control word and the corresponding binary pulse width modulated control signal. Error detecting and correcting codes could be used for the control word if power is available for the more complex decoder in the APE. At the present level of 100mW available power to an APE, these desired coding schemes are not incorporated into the machine.

Yiu Wo



NUMBERS IN CIRCLE INDICATE THE PIN NUMBER OF IC

Figure 2. Circuit of 32-bit Control Code Generator.



(a) A CONTROL WORD

(b) THE CORRESPONDING PULSE WIDTH MODULATED SIGNAL

Figure 3. Control Word and Corresponding Pulse Width Modulated Signal.

1.3 PENTECOST (Project No. 31)

1.3.1 Monitor Modification

The black and white Conrac monitor is being modified to permit installation of the Penetron tube. This is an RCA development type dual color phosphor screen, 17"-rectangular, 70° magnetic deflection. The screen of the Penetron consists of a white transparent layer of phosphor over which a red layer of phosphor is deposited. At a screen voltage of 11KV the electrons excite the red layer only. At 16KV the electrons mostly penetrate through the red layer and excite the white layer of phosphor producing a white approximately equivalent to CIE illuminant "C". This display tube employs a single high-voltage electrostatic focus electron gun and the focus voltage has to be switched from 1450-2100 to 1850-2700. Likewise the vertical sweep amplitude and horizontal sweep amplitude have to be switched by a factor of $\sqrt{16/11} \simeq 1{\cdot}2$. The brightness level of white phosphor is twice that of red phosphor. So with the change of screen voltage, the video gain of the amplifier and the cathode to grid bias voltage have to be switched.

1.3.2 Screen Voltage

It is planned that the Conrac monitor screen voltage, which is regulated at about 19KV, will be switched to provide the screen voltage for the Penetron. A high voltage switch, fast enough to change the screen voltage during the vertical retrace time, is being built.

1.3.3 Focus Voltage

The focus voltage supply of about 4KV will be obtained from the 19KV monitor high voltage by a resistance divider network. This voltage also will be switched by a high voltage switch. In the Conrac monitor the screen voltage is about 400V, to which a dynamic focus correction voltage will be added.

1.3.4 Horizontal Deflection

Since we are using the high voltage output derived from the horizontal sweep circuit, we cannot employ any potentiometer switch network at the input or output of the horizontal output tube, because this would not only change the horizontal sweep amplitude but also would vary the high voltage output. So we will place a variable-core coil, similar to the width coil, across the yoke coil. This coil will be switched in and out of the yoke circuit to switch current amplitude in the horizontal yoke.

1.3.5 Vertical Deflection

The gain of the vertical sweep output tube can be changed by switching a resistance input network.

1.3.6 Video Circuit

A switched resistance network will be used to vary the output signal from the contrast control. The grid to cathode bias voltage will be switched by using a similar resistance network at the brightness control.

A block diagram of the Pentecost receiver is shown in Figure 1.

1.3.7 PENTECOST Camera

A red-extended Plumbicon tube (Amperex 16 x QRIG) has been received. This will be installed in GE T-26 camera with a few minor modifications. A filter wheel consisting of alternate red and green filters will be driven by a synchronous motor. Elaborate motor drive circuitry will be needed to synchronize the wheel with the vertical fields. A photosensor detects the position of the wheel and compares this with the vertical sync and makes a correction so as to drive the motor synchronously. The field indicator sync is obtained at this point. A block diagram of the camera system is shown in Figure 2.

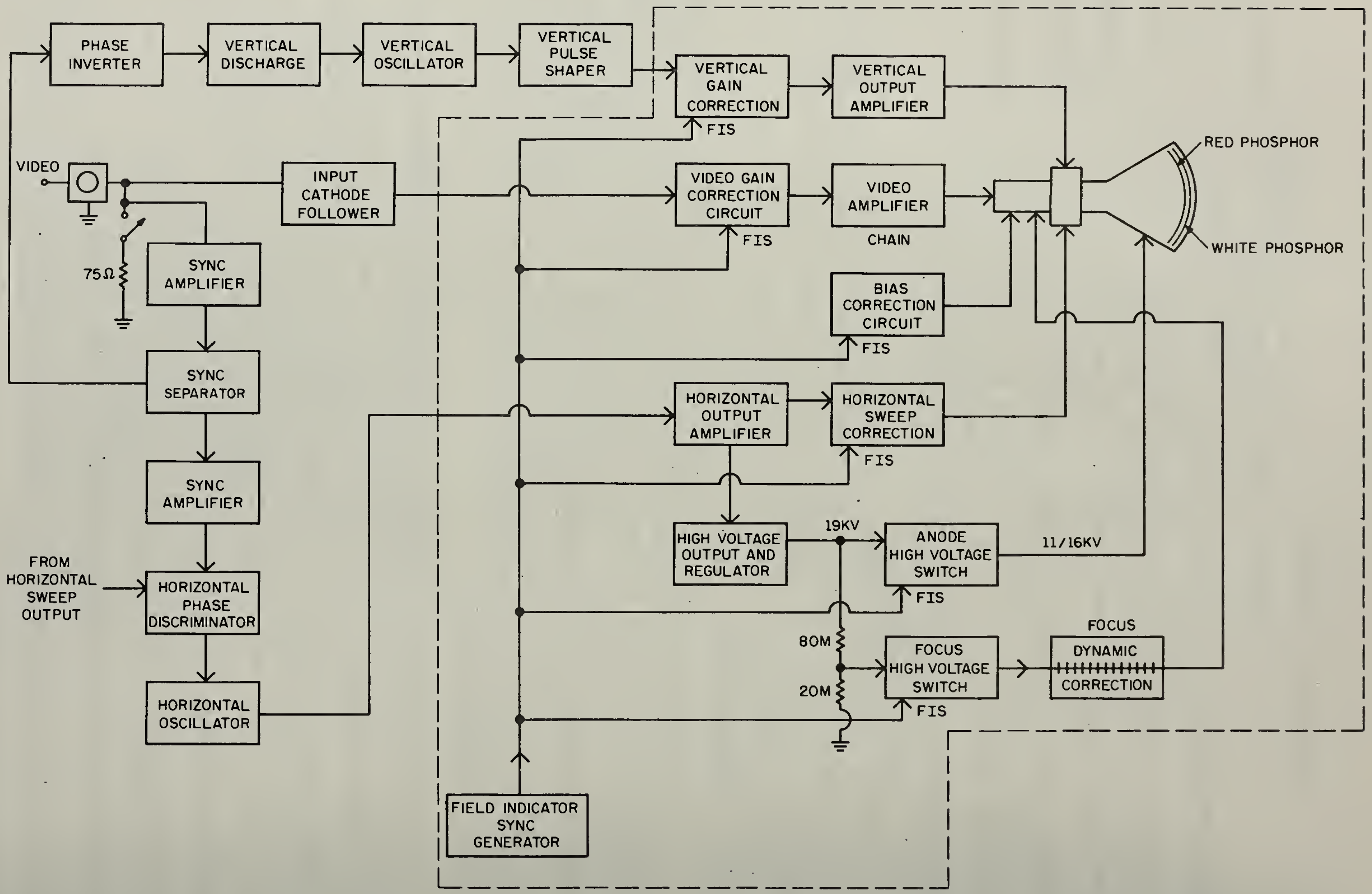


Figure 1. PENTECOST Receiver

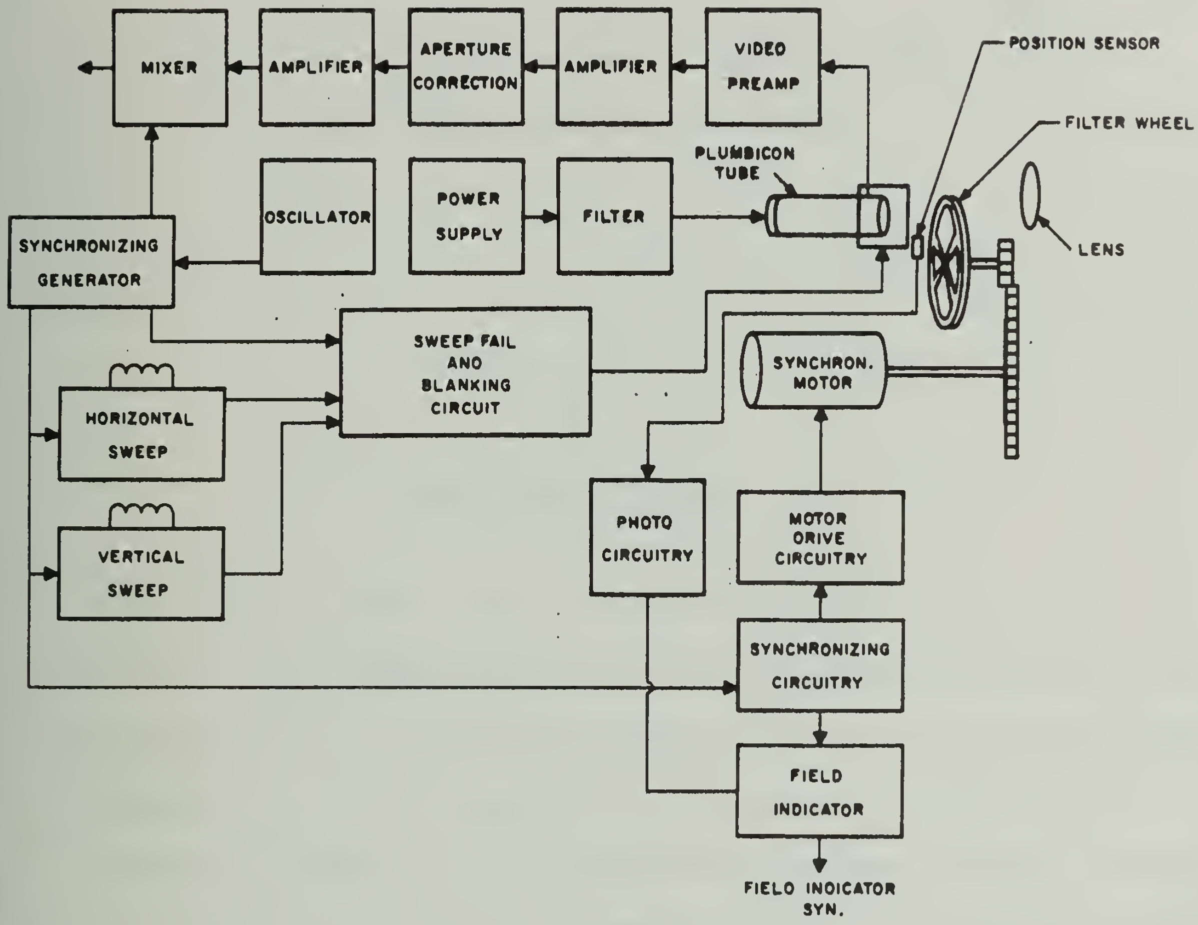


Figure 2. PENTECOST Color Camera Camera with Filter Wheel

The possibility of using a voltage controlled electronic filter instead of the filter wheel is being looked into.

G. Panigrahi

1.4 Ergodic (Project No. 39)

1.4.1 Outline

This project stems from two previous and completed projects done by this group: BUM and RASCEL. Both of these were concerned with representing numbers by a random process and doing arithmetic operations on these numbers. In RASCEL numbers were represented by a random time sequence of binary voltage levels ("1" and "0") along a wire. Thus, it may be said that this scheme has a time average of a wire being at "1" or

$$\text{time average} = \frac{\text{amount of time that wire is at "1"}}{\text{total observation time}}$$

Now consider a bundle of wires in which every wire has a random sequence of binary voltage levels. If the time average of any wire is the same as the space average of the bundle, then we call this an ergodic bundle.

The goal of this project is to build a system using an "ergodic generator" that will consist of checkpoints; i.e., a unit that will add, subtract, multiply and divide the numbers represented by two ergodic bundles.

1.4.2 Project Status

Since this project has just started, progress has been at a minimum. Design work is now being done for a particular ergodic generator. The major concern will be the ergodic generator. Two techniques will be considered for the ergodic generator and each will be tested and evaluated.

Jim Cutler

1.5 Telemaze (Project No. 41)

1.5.1 Introduction

It is well known that feedback for on-line operation is only useful if the delay in the loop is short enough. A typical example where feedback is nearly useless is the control - from earth - of an exploratory vehicle on a remote planet: either we have to go so slowly that the vehicle hardly moves in "one feedback time", or we must risk destruction by unknown obstacles. It is now possible to imagine a system in which all that is known about the environment of the vehicle (i.e. an environmental "model") is stored in a computer on earth. Guidance is then obtained by steering an "image" of the vehicle through the model. Two cases arise: either the real vehicle (which follows with some delay the "image" in the model on earth) does not pick up any unexpected information - e.g. the position of an obstacle. Then we can happily continue through our "imaginary world" in the computer. Or, and this is the important case, an obstacle does occur. The vehicle then comes to a standstill, sends out a "hold it" signal to earth and updates the model in the computer. The driver will go back to the position before the "hold it" signal came in and continue his "excursion" through the updated model. The important point is that in general he will have a pretty good model right from the beginning (previous photographs, etc.) and that the update process will only happen relatively rarely. Thus, the vehicle can proceed at a reasonable speed, losing only a minimum amount of time due to feedback delays.

The goal of TELEMAZE is to design a simple system with an adjustable feedback delay, obtained by encoding the information on a magnetic tape loop, the latter running at a speed the operator can select. To come close to the "vehicle problem" described above, we use a 16 x 16 array of squares, a "black" square corresponding to an obstacle, a "white" one to a free path. The arbitrarily chosen (and perhaps modifiable - while running!) "maze" of occupied and unoccupied squares has the "vehicle" projected by a flying spot scanner. This is a point obtained from the "image" (inscribed by an X, Y position tablet) after due delay. The transfer is via two buffers, as shown in Figure 1. Clearly it is possible for the Remote Buffer to be quite a bit behind the "image" positions stored in the Local Buffer.

1.5.2 X, Y Coordinate Control Logic

The basic function of the X, Y coordinate processor, Figure 2, is to take trajectory information from the input tablet and store it in a buffer until it can be transmitted to the remote vehicle. The tablet contains the logic necessary to resolve the path into discrete X and Y coordinates, by energizing one of 16 X, Y wires. This information is then encoded into a 4-bit binary code for each axis. The displacement logic will compare the previously processed word (8 bits) with any non-zero requests. This is necessary to ensure a continuous trajectory. Having successfully passed as a new point, a command is then given to write the new word into the large buffer. To accomplish this, the address of the last word entered into the buffer is found in the write address counter. It will now count up by one address (the buffer is implemented as a circular memory) and place the new word into the buffer. Reading this information back out of the large buffer is completed by the "Read" logic. After a word is read out of the buffer, the read buffer counter will count up by one address. Reading this information out will be slower than writing because of parallel to serial conversion

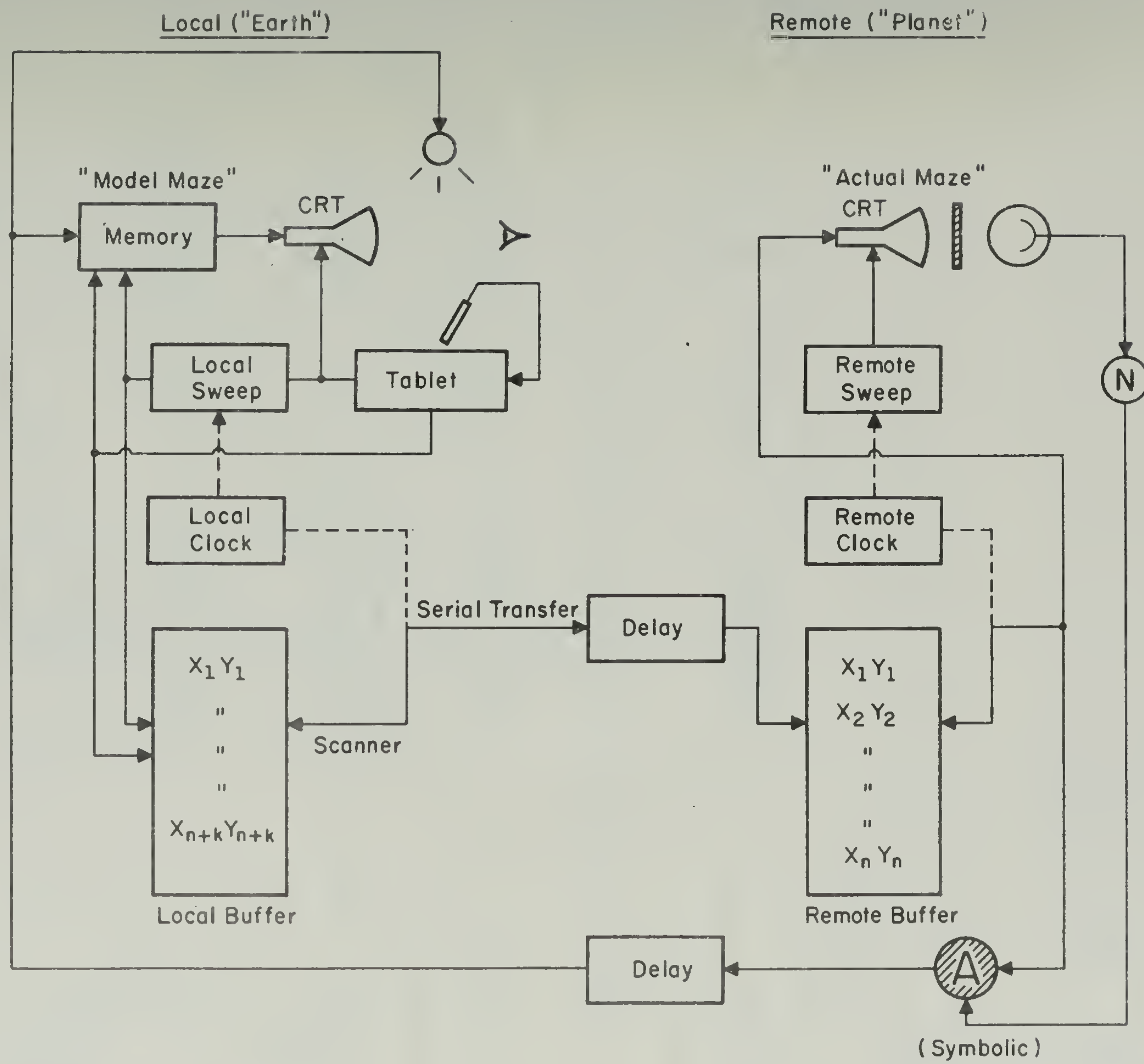


Figure 1. Telemaze Model Feedback System

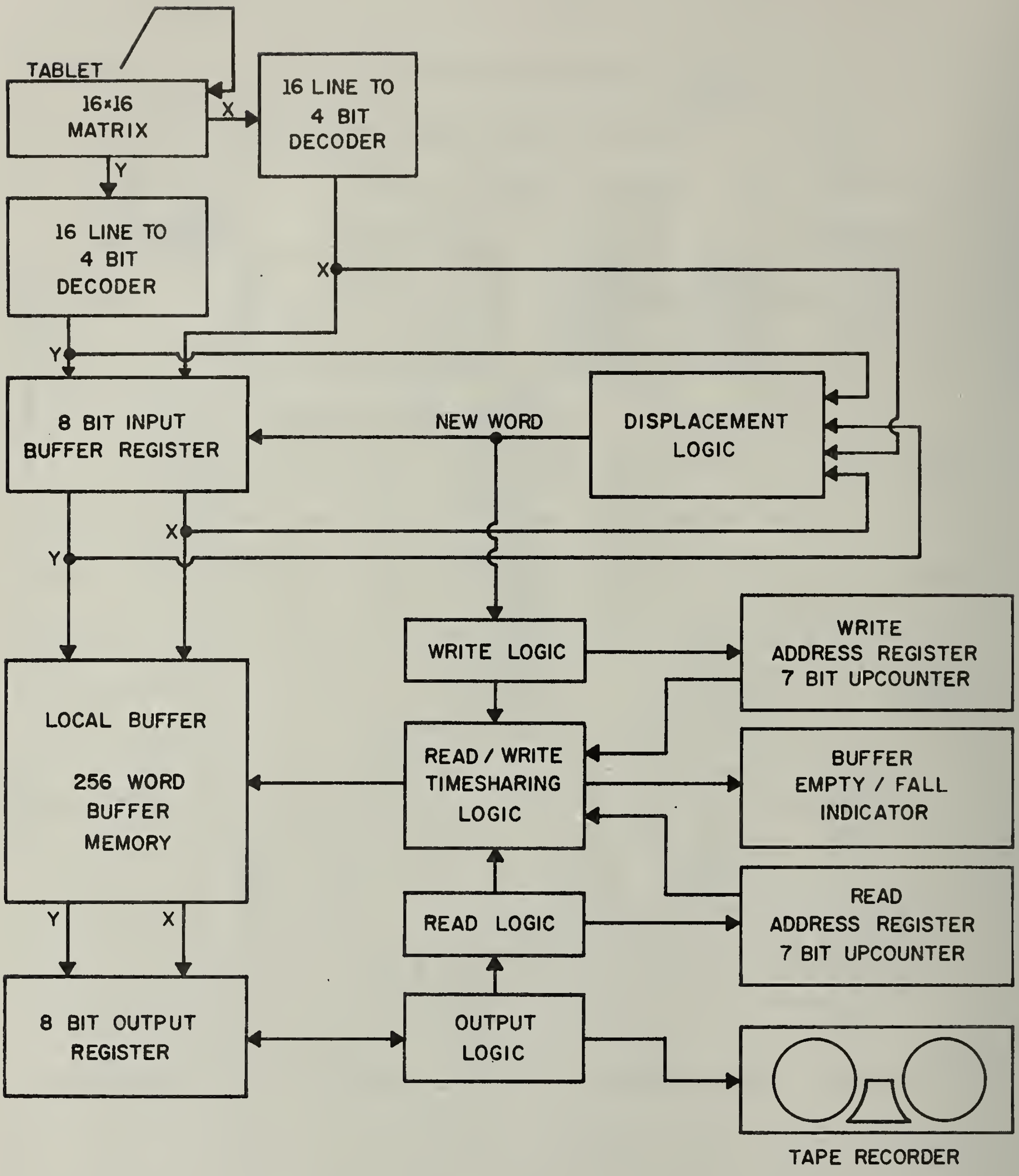


Figure 2. Block Diagram of (X, Y) Coordinate Control Logic

needed for transmission. To allow for the difference in these time intervals, additional logic for timesharing the buffer has been added. If both read and write circuitry are active, control is passed alternately to each as long as they both request. If there are no new points from the tablet to be located in the buffer, the reading circuitry will have full control and vice versa.

Data points read out of the buffer are then processed and applied to magnetic tape. By controlling the tape speed, transmission delay time can be varied.

Edward Pott

2. HARDWARE SYSTEMS RESEARCH

(Supported in part by the Atomic Energy Commission under Contract US AEC AT(11-1) 1469, W. J. Poppelbaum, Principal Investigator.)

Summary

Doug Sand's OLFT report deals with preliminary experiments in operating the crystal transducer at reduced temperature. Don Hanson has some new circuit diagrams for the Tricolor Cartograph. Larry Wallman discusses the problems of dynamic scan correction for BLAST, the stereo TV display project. Semantrix has the design of its block detection circuitry almost complete. Trevor Mudge also describes progress on the construction of the arm and the computer interface logic. Dick Blandford writes about the fiber optics and linear photocell array which LINDA uses to determine the identity of polygonal figures. Shiv Verma and Steve Whiteside are putting the finishing touches on Stereomatrix; their reports deal with the transformer, coefficient generator and display. The Scantrix project entails line-at-a-time storage for a LED, large screen display. Sik Yuen gives details of the A/D converter for incoming video information. Finally, Debasish Bose writes about preliminary work on FROG. This is a new project under the supervision of Professor Ray which aims to design an autonomous mini-robot that can adapt to its environment.

M. Faiman (ed.)

2.1 OLFT (Project No. 12)

2.1.1 Operation of Intermediate Cooled System

Exploratory operation of the intermediate cooled Pockels tube continues, with testing of multiple-frame writing and manual pulsed erasure at substrate temperatures down to -30°C. The apparent charge-decay time-constants are about as anticipated, with a longitudinal τ on the order of two minutes at -30°C. The transverse, or surface τ appeared to be somewhat shorter, but could not be estimated due to image resolution limitations. The low resolution was caused by electronic components: the sweep circuits again contained excessive noise, and the isolation amplifier could not compensate for the large dc-level change caused by frame-gating. The effects of these and other electronic sources of image degradation have been minimized, and testing at lower temperatures will resume.

2.1.2 Optical System Design

Some of the detailed design for the required optical components has been completed. However, a fundamental modification is being considered which will simplify system construction and alignment. The change is the use of a parabolic mirror, rather than a lens, to form the Fourier transform. In addition, due to the high cost of commercial high-resolution positioning devices, several alternatives suitable for in-house construction are being considered, such as differential screws and piezoelectric translators.

Doug Sand

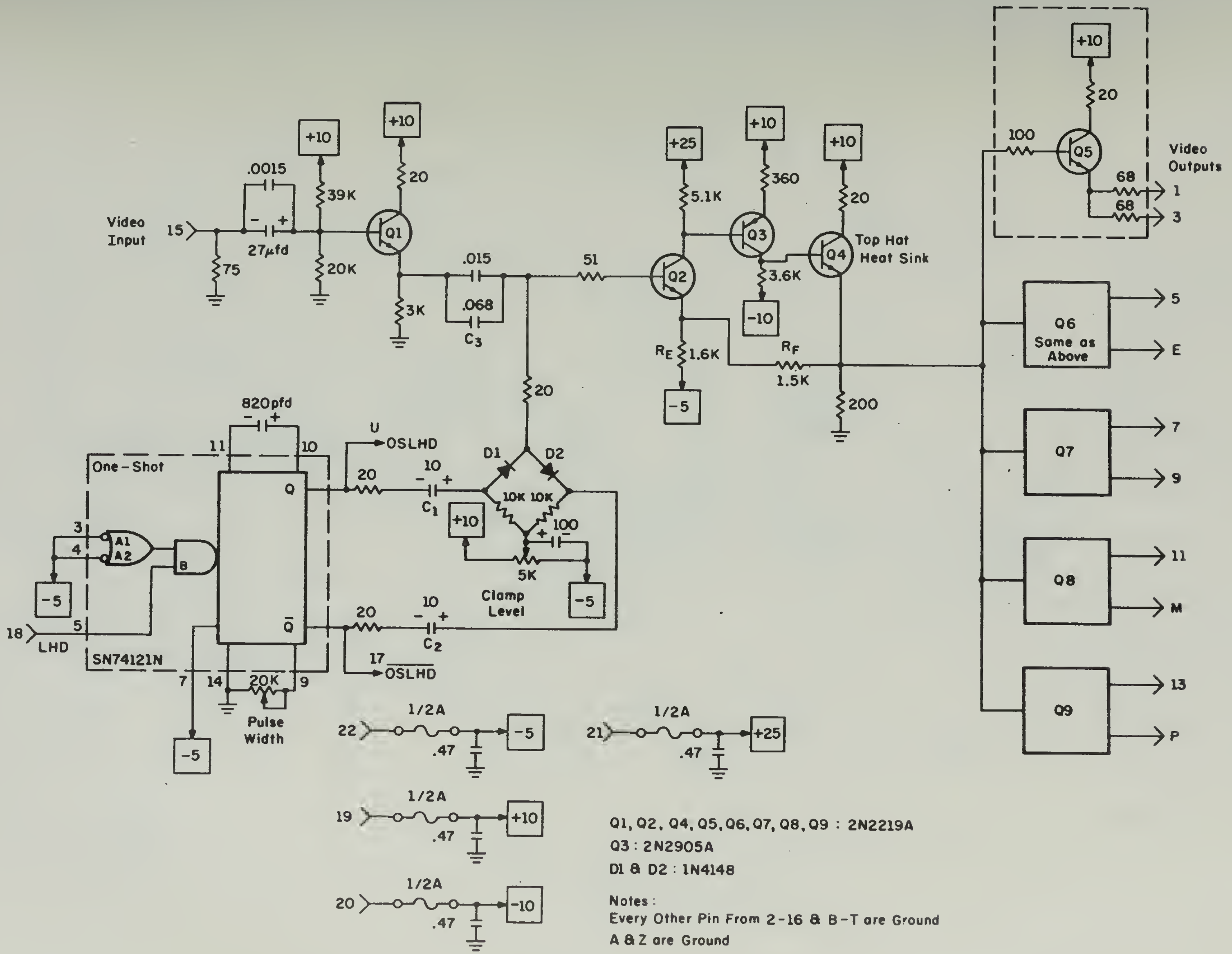
2.2 Tricolor Cartograph (Project No. 16)

2.2.1 Circuits

Recent work has been devoted to designing and building special circuits for the modified Tricolor Cartograph. These are as follows:

Figure 1.	Video Distribution Amplifier
Figure 2.	Horizontal Differentiator
Figure 3.	Synchronous Chopper
Figure 4.	Corning Delay Line Circuit
Figure 5.	Vertical Differentiator
Figure 6.	Video Gate

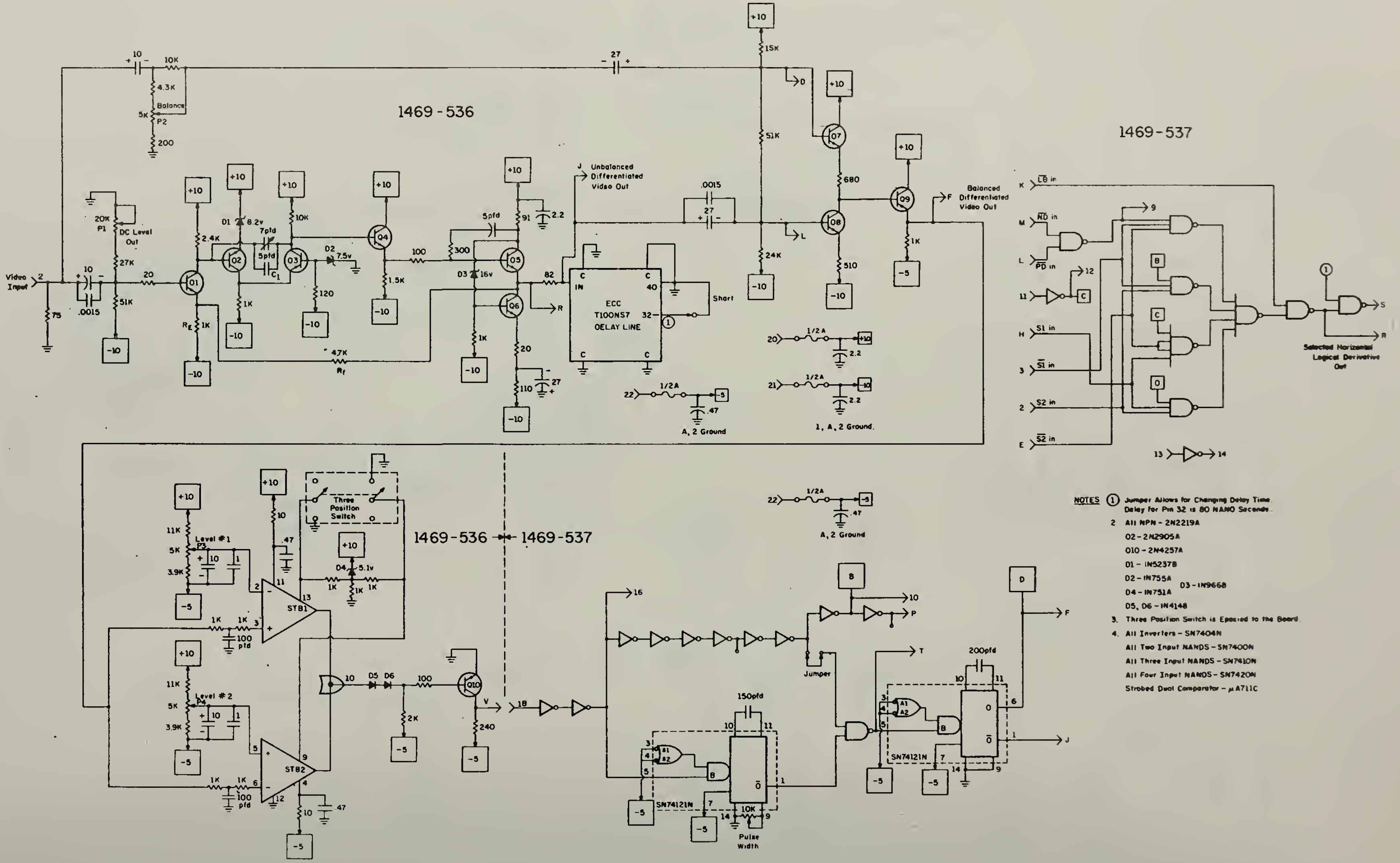
Don Hanson



Q1, Q2, Q4, Q5, Q6, Q7, Q8, Q9 : 2N2219A
Q3 : 2N2905A
D1 & D2 : 1N4148

Notes:
Every Other Pin From 2-16 & B-T are Ground
A & Z are Ground

Figure 1. Video Distribution Amplifier



NOTES ① Jumper Allows for Changing Delay Time. Delay for Pin 32 is 80 NANO Seconds.

2 All NPN - 2N2219A

Q2 - 2N2905A

Q10 - 2N4257A

D1 - IN5237B

D2 - IN755A

D4 - IN751A

D3 - IN9668

D5, D6 - IN4148

3. Three Position Switch is Epoxied to the Board.

4. All Inverters - SN7404N

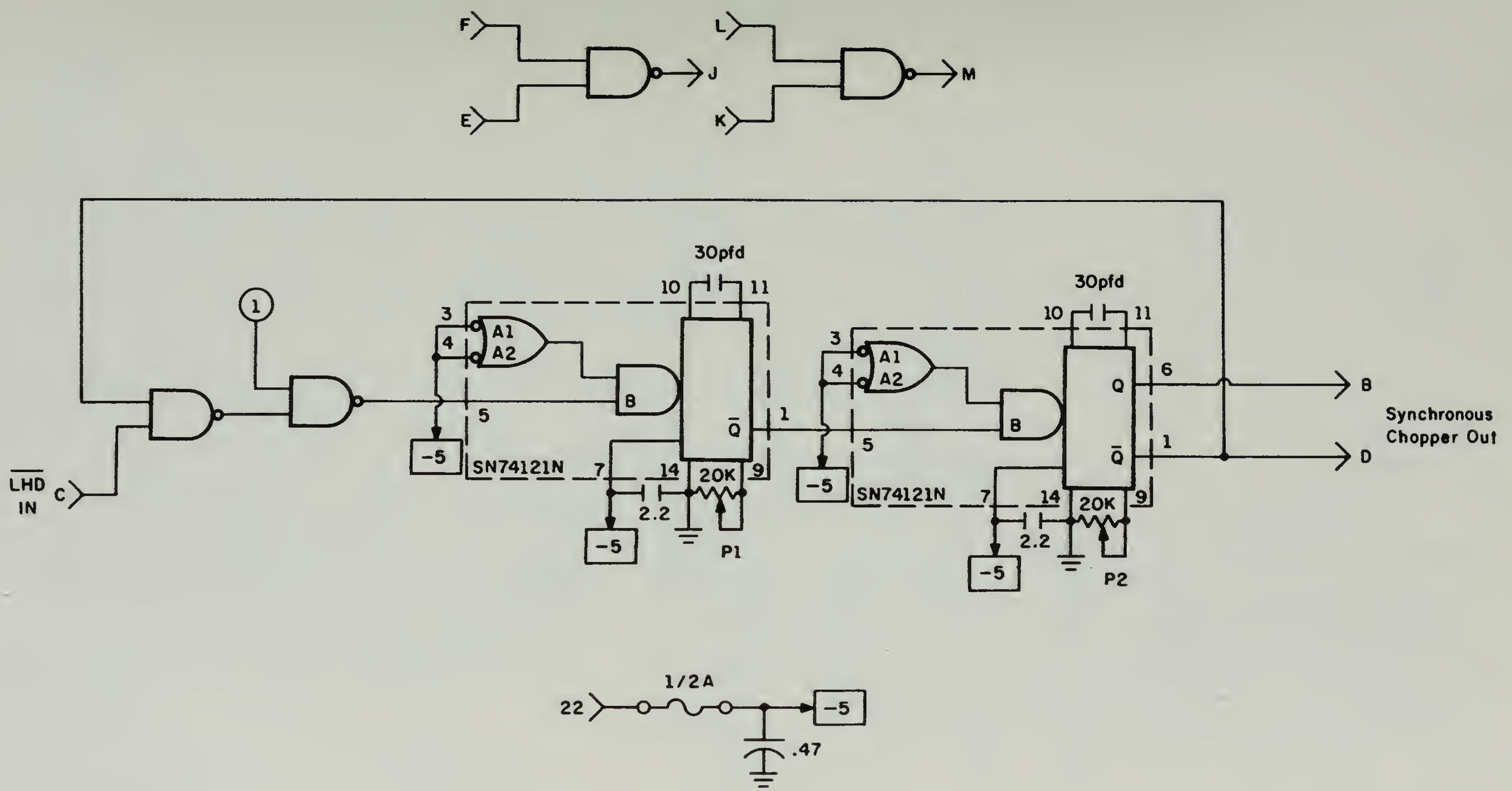
All Two Input NANDS - SN7400N

All Three Input NANDS - SN7410N

All Four Input NANDS - SN7420N

Strobed Dual Comparator - μA711C

Figure 2. Horizontal Differentiator



NOTES: 1. P1 and P2 are Frequency Controls
2. NAND Gates : SN7400N
3. Pin Z is Ground
4. Circuit Built on 1469-218 Card

Figure 3. Synchronous Chopper

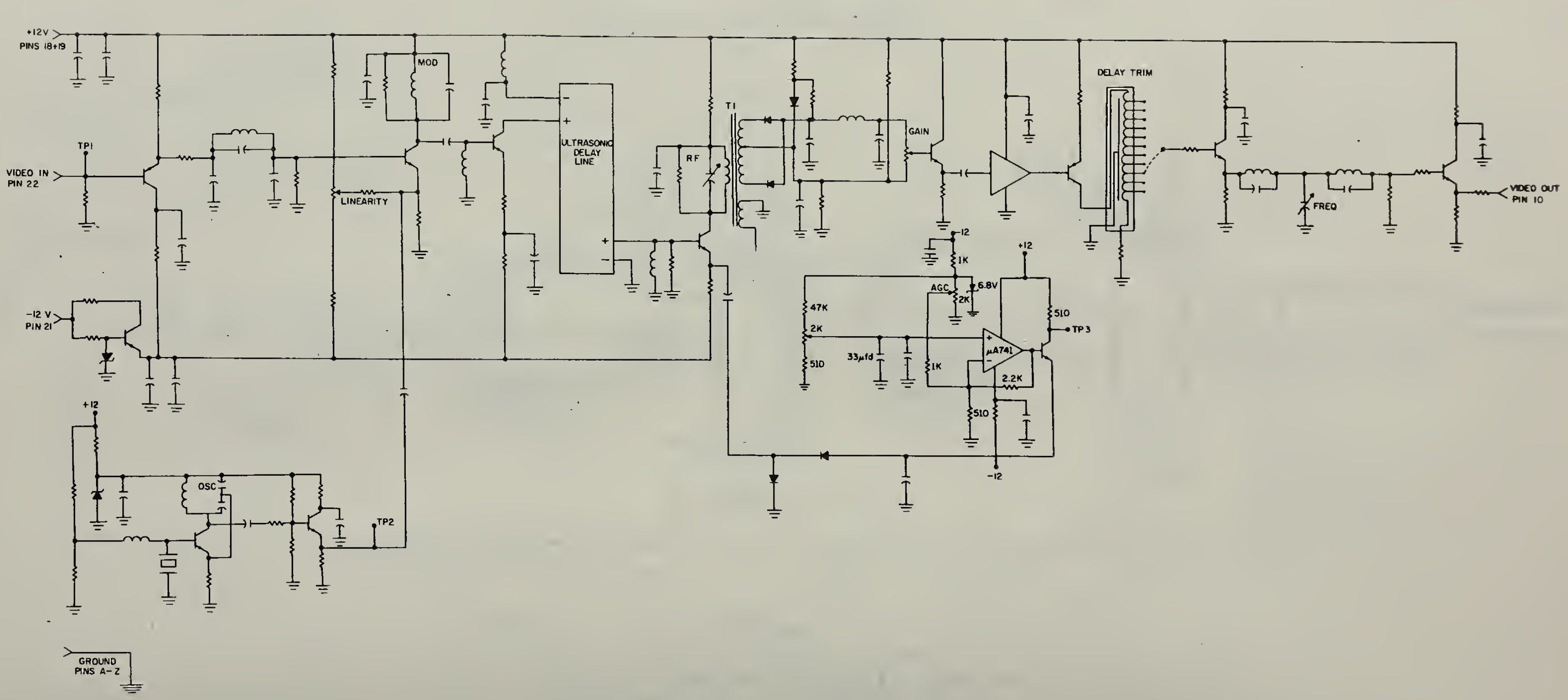
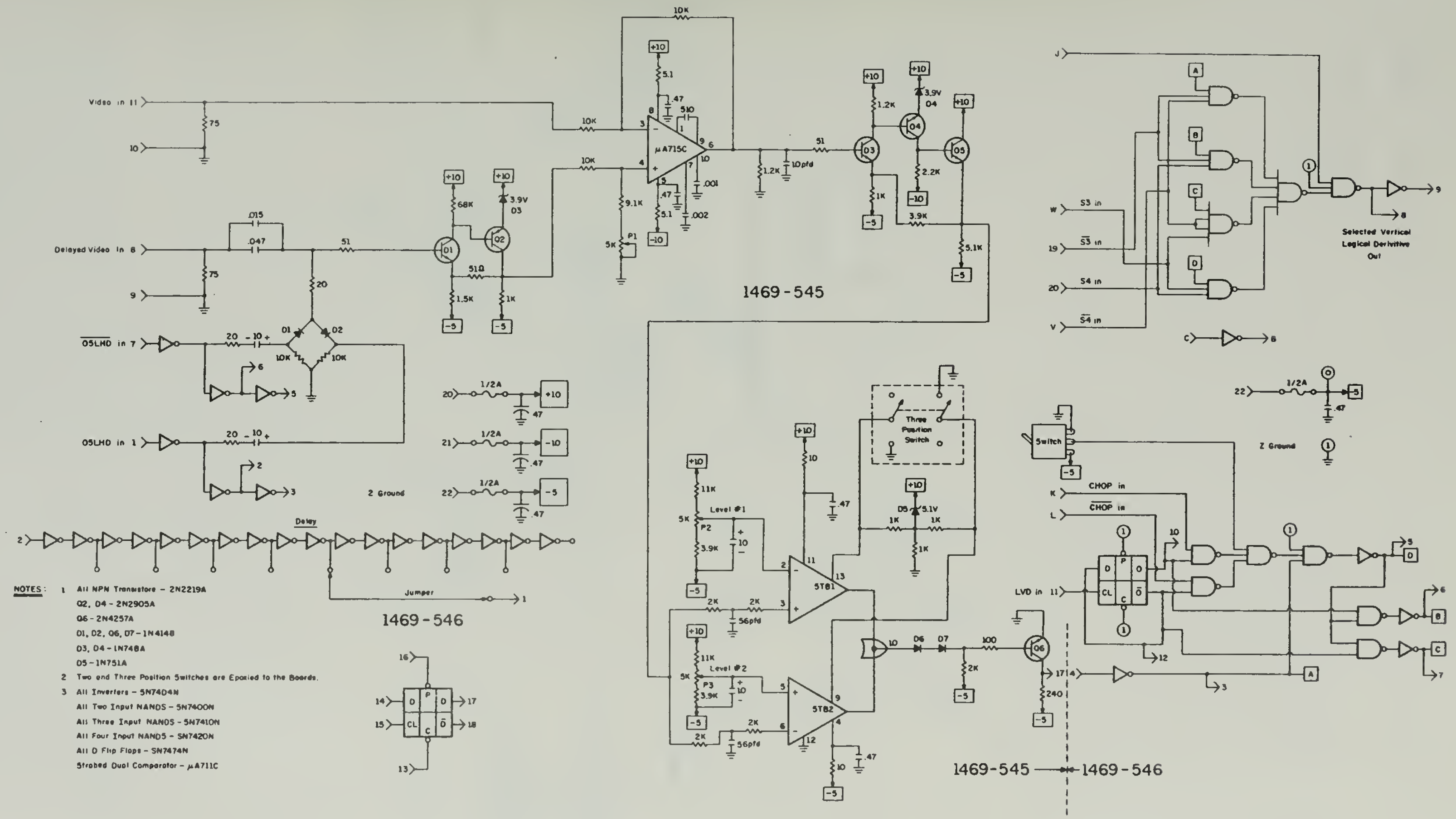


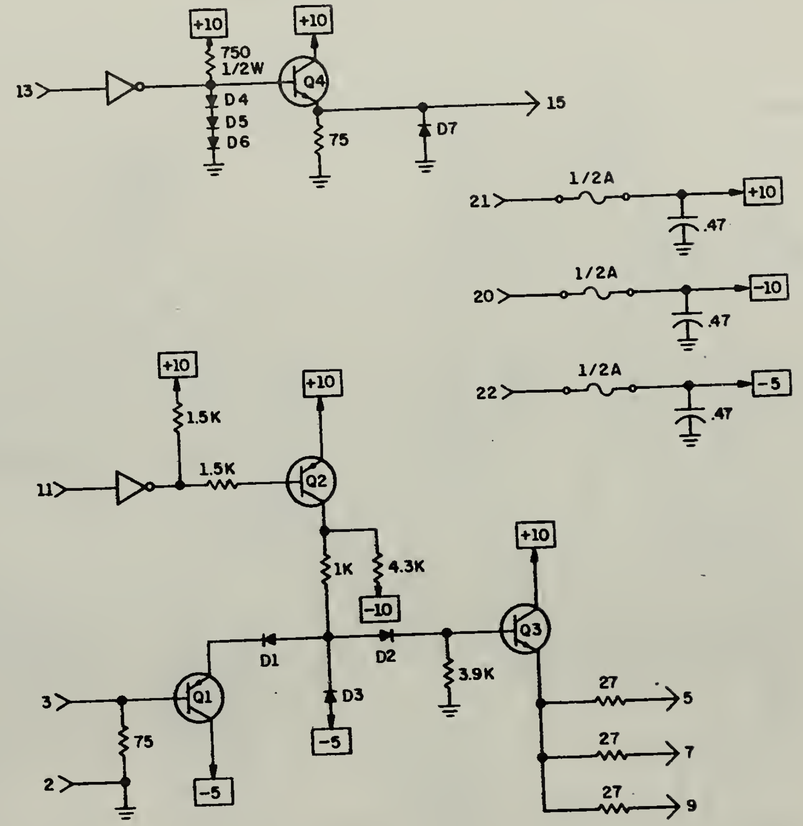
Figure 4. Corning Delay Line Circuit



NOTES:
1 All NPN Transistors – 2N2219A
Q2, Q4 – 2N2905A
Q6 – 2N4257A
D1, D2, D6, D7 – 1N4148
D3, D4 – 1N748A
D5 – 1N751A
2 Two and Three Position Switches are Epoxied to the Boards.
3 All Inverters – SN7404N
All Two Input NANDS – SN7400N
All Three Input NANDS – SN7410N
All Four Input NANDS – SN7420N
All D Flip Flops – SN7474N
Strobed Dual Comparator – µA711C

Figure 5. Vertical Differentiator

1469-554



NOTES

1. Q1: 2N2905A
   Q2: 2N3905
   Q3,Q4: 2N2219A
   D1 - 1N995
   D2,D3 - 1N4151
   D4,D5,D6 - 1N4148
2. PINS 1,2,4,6,8,10,16 are ground.
3. INVERTER: SN7406N

Figure 6. Video Gate.

2.3 BLAST (Project No. 19)

2.3.1 Screen Signal

It was reported last time that the internal connection to the interwoven conducting stripes had broken down. It was conjectured then that this happened because of the "sandwich" type connection made to the nickel conductors on the lenticular screen. The tube manufacturer was contacted during this quarter and he indeed verified this conjecture. The precaution they had taken to prevent this was to wrap aluminum foil around the connection to give a greater contact area.

2.3.2 Dynamic Scan Correction

The scan correction required is different from the normal linearity correction used in cathode ray tube displays. The latter ensures that the raster is rectangular when viewed from the front. This correction is required since, with a linear yoke current, the raster will appear rectangular when viewed from the front only if the tube screen is the cap of a sphere concentric with the deflection center. Since most tubes have screens flatter than this, a correction is required. This is shown in Figure 1. For BLAST the requirement is that the horizontal scanning speed be constant since the video chopper-mixer frequency is constant. Calculation shows that the horizontal yoke current for a constant velocity scan is given by

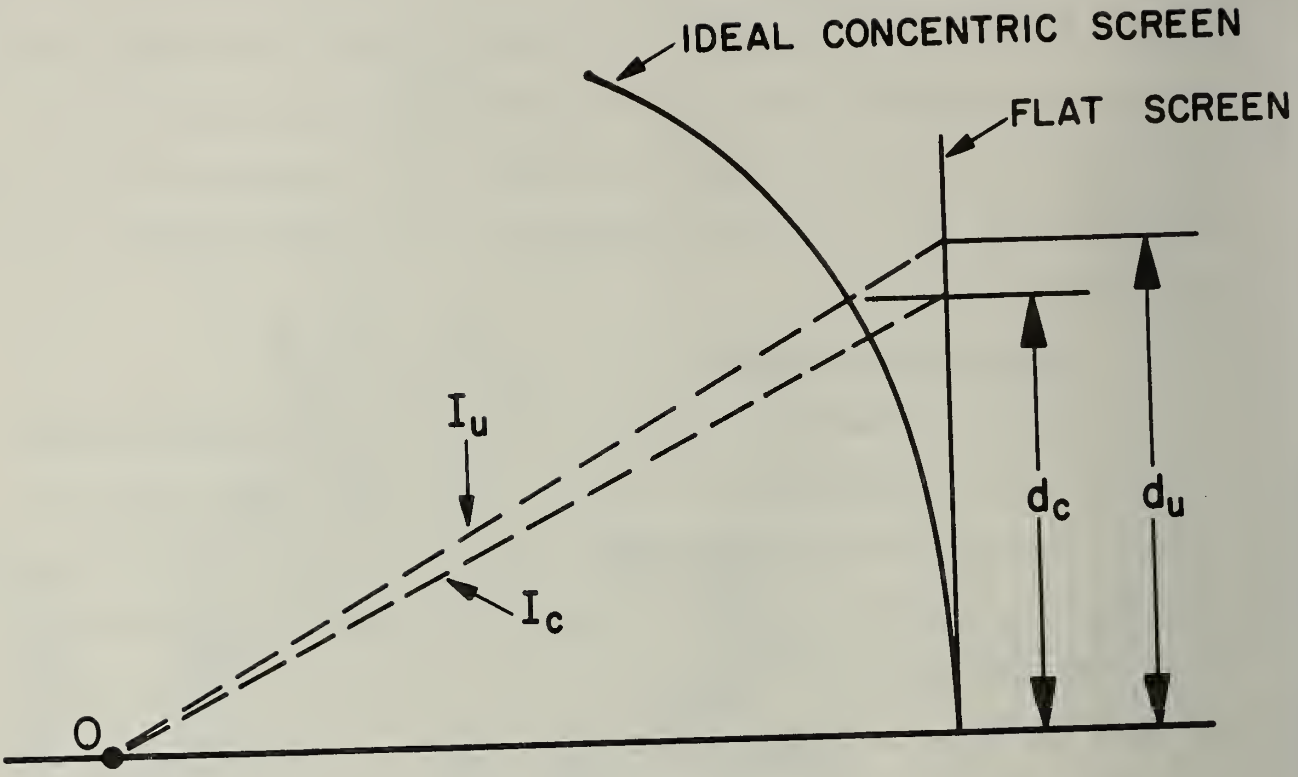
$$I_H = k_1 \sin [(k_2 \cos \varphi) t]$$

where k_1 and k_2 are constants, φ is the vertical deflection angle and t is time. Also, φ is related to the vertical deflection current I_V by

$$\sin\varphi = k_V I_V$$

with k_V constant. But since φ is small, its sine can be well enough approximated by the argument, so that the horizontal current may be written

$$I_H = k_1 \sin [(k_2 \cos k_V I_V) t]$$



d_u : uncorrected deflection

d_c : corrected deflection, projection of deflection on concentric screen onto flat screen

I_u : uncorrected deflection

I_c : corrected deflection

O : Deflection center

Figure 1. Normal Linearity Correction

In designing the yoke driver several things have to be considered. The driver must be linear so that the current can be controlled precisely. This precludes using a flyback type of drive circuit since control is not maintained over the entire cycle. The severest requirements occur during retrace since only 10 microseconds are available to retrace the yoke from plus 300mA to minus 300mA. The horizontal yoke has an inductance of approximately 10mH. This would require at least -600v to retrace in 10 microseconds. Therefore it was decided to use external dynamic correction coils which have a lower inductance and a lower current requirement since the horizontal yoke would do the major deflection. A feedback type of coil driver is used and has been laid out on printed circuit cards. Some parasitic oscillations have occured in the printed circuit cards which did not occur in the breadboard model. Currently these are being eliminated.

Larry Wallman

2.4 Semantrix (Project No. 28)

2.4.1 Summary of Project

Semantrix is intended to serve as an I/O device to a general purpose computer. The combined system may then be used to investigate the feasibility of the "World Model" approach to imbedding intelligent behavior in an electronic system.

Semantrix has a flat table top (3' x 3') upon which colored blocks may be placed at random. Up to 64 of these blocks may be used simultaneously and their X-Y coordinates are transmitted to either a computer or a TTY upon receipt of a suitable request. This is accomplished automatically at speeds compatible with either a TTY or a typical computer. The center of each block can be located to within a radius of 0.4" of the true center. Together with this Semantrix has an electromechanical arm which can move blocks to prescribed points on the board. These points and the blocks to be moved can be specified by the TTY or the computer. Hence, either can request the position of any block and is also capable of altering its position.

The first investigation to be undertaken is to test the "correctness" of a world model for landscape scenes. Given a random array of colored blocks (blue, green, white, red, etc.) an 8 x 8 picture is synthesized by the world model (a Fortran program in the computer) through Semantrix.

2.4.2 Project Status:

The project subdivides into 3 main areas:

1. Block location

2. Electro-mechanical arm

3. Logical interface with both a computer and a TTY.

1. Most of the circuitry associated with this part completed in its final form. and is being installed in the table.

2. With the recent arrival of a pair of suitable torque-proportional-to-input voltage motors, construction has started and is expected to be complete by the end of the year.

3. Most of the logic has been fabricated, however debugging must await the arrival of the TTY.

Trevor Mudge

2.5 LINDA (Project No. 28)

2.5.1 Summary of the Project

LINDA is a attempt to build a LINe Drawing Analyzer capable of identifying a small number of simple line drawings. A given drawing is projected on a screen by means of a flying spot scanner. Edge information is extracted by means of photocells. The photocells are arranged in a linear array and discontinuities in the figure are of prime interest (see previous reports for an explanation of the details of the procedure used).

2.5.2 Project Status

During the past quarter work has been directed at building a linear array having a suitable resolution. An array has been built which incorporates fiber optic light pipes (see Figure 1) and dual element photocells. The light pipes allow closer spacing than could be accomplished with photocells alone. There are 52 light pipes in about 2 3/4 inches. The ends of the pipes touch the screen of the CRT. When a figure is swept past the light pipes the ordinate sum is taken and differentiated twice to produce pulses corresponding to discontinuities in the figure. The amplifier cards for the photocells are being built and at this time a complete test of the arrangement has not been made. Preliminary tests using only 10 photocells on a similar system indicate that the present arrangement will be satisfactory.

2.5.3 Future Work

In the coming quarter tests will be carried out to determine the resolution of the photocell light pipe arrangement and improvements will be made if necessary.

Work will also be directed at developing the logic for a system which can scan a pattern, separate it into its simple parts, and sweep these parts, one by one, across the linear array for identification. As a simple example, consider the "house" shown in Figure 2. The system logic will first determine that there are two simple patterns in this figure. It will then color one pattern so that it is all one color rather than a simple line drawing. The second pattern will then be erased and the solid colored figure

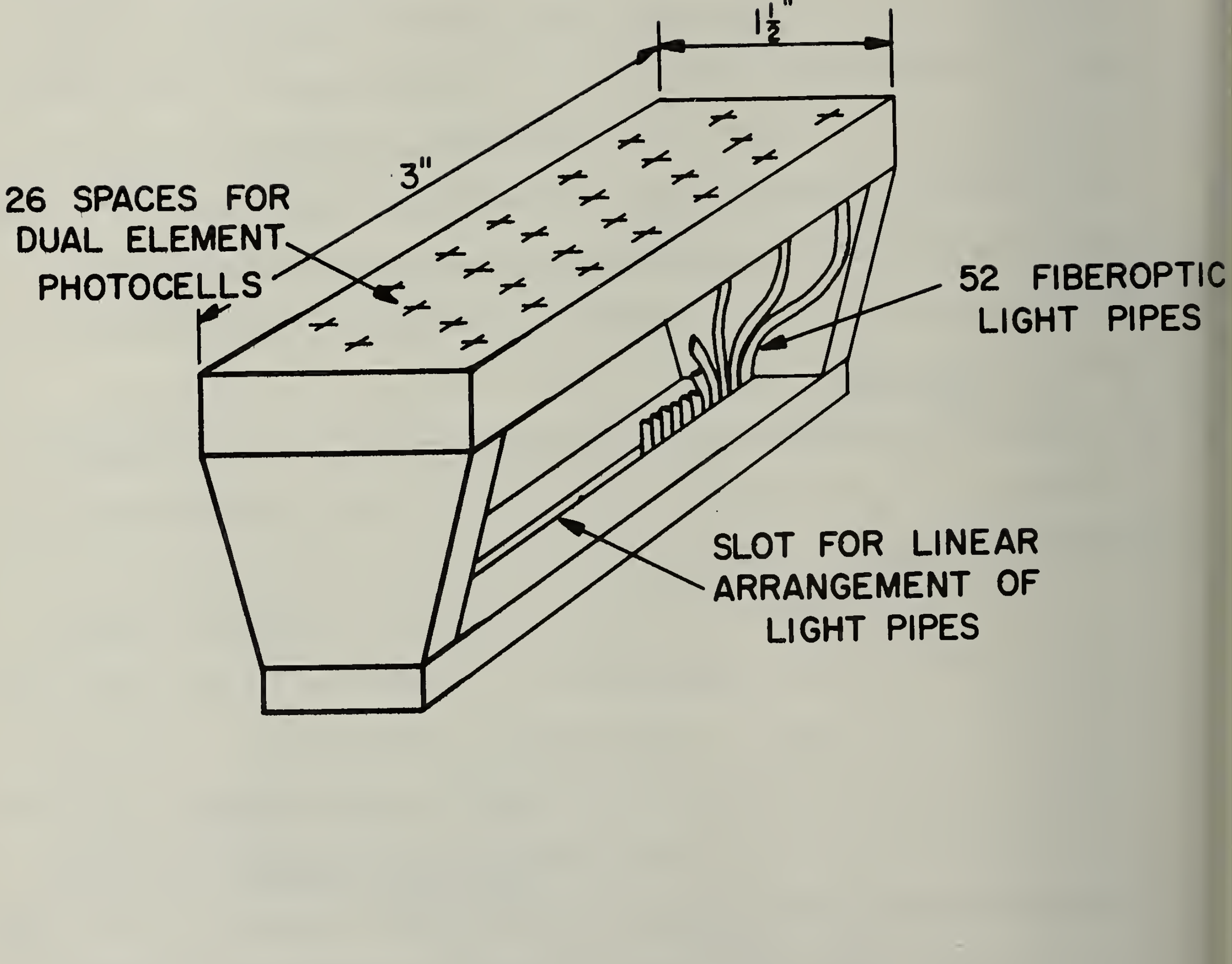

1½"
3"
26 SPACES FOR DUAL ELEMENT PHOTOCELLS
52 FIBEROPTIC LIGHT PIPES
SLOT FOR LINEAR ARRANGEMENT OF LIGHT PIPES

Figure 1. Photocell Arrangement Showing Fiber Optic Light Pipes for Linear Array

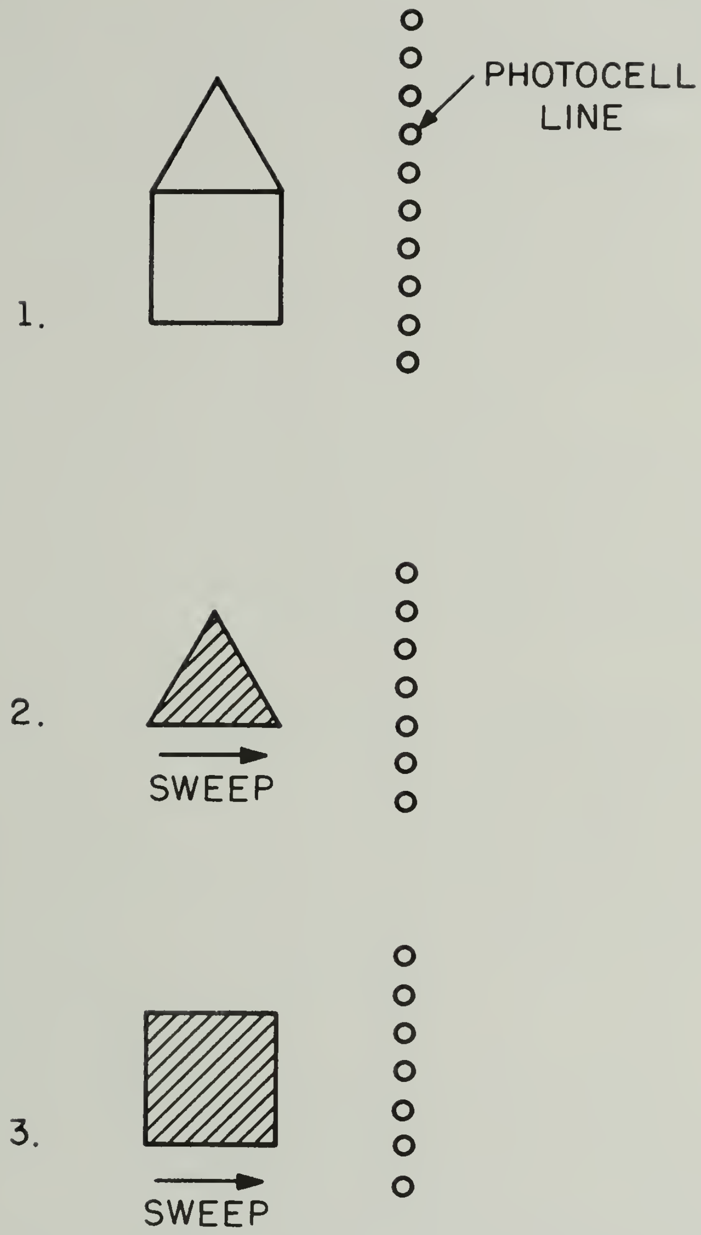


Figure 2. Steps in Identification of a "House"

will be swept across the linear array for identification. It will then return this identified pattern, erase it, color in the second pattern and identify it. A triangle over a square in this figure will be called a "house". Many other figures are possible. At this time a vocabulary of about 25 figures are being considered. This will probably be expanded later.

Dick Blandford

2.6 Stereomatrix (Project No. 30)

2.6.1 Transformer

The transformer has been in operation since July. Rotation and translation coefficients from the coefficient generator have been supplied to the coordinate transformer unit and the X, Y, Z outputs of PAGAN have been used as the input to the system. A left eye view only (X_L, Y) of the PAGAN pattern obtained from the transformer was displayed on the CRT. The change in perspective could be observed by rotating the pattern about the coordinate axes or by translating it or by changing the observer's position.

Thus, all the features of the transformer have been very well tested. The scaling circuit has been modified and a new card has been made. The gain at the output of the system has been adjusted to make the transformer compatible with the input to the deflectors.

The driver and receiver cards needed to interface the transformer with the PDP-8I have been wired and are being tested. Picture and display control logic needed to communicate with the PDP-8 are being wired.

Thus, it is hoped that the transformer will be interfaced and tested with software from the PDP-8 in the next quarter.

2.6.2 Coefficient Generator

Dick Cheng has finished his thesis and left. However, there is a lot to be accomplished in the coefficient generator. It is expected that we will soon have the wiring and logic diagram of the coefficient generator checked and that the system will be working shortly.

Logic design for the digital read out of all twelve coefficients has been completed and cards have been laid out and tested.

Shiv Verma

2.6.3 The Display

The rear projection screen has arrived and does give a reasonable extinction ratio (of proper to spurious polarization component). The extinction is not quite as good as with the aluminized lenticular front projection screen, but it is quite acceptable. The shop is building a wooden frame for the screen to bolt on the viewing desk. The rear projection screen seems to allow much more front lighting without "washing out" the picture. Some figures could easily be viewed in the normally lighted room. With the good quality, coated optics that have been ordered there is some hope that the system will perform well enough to be used with normal room lighting.

The deflector has just been returned from the vendor and remains to be tested. The frequency instability problem was reportedly in the wideband power amplifier.

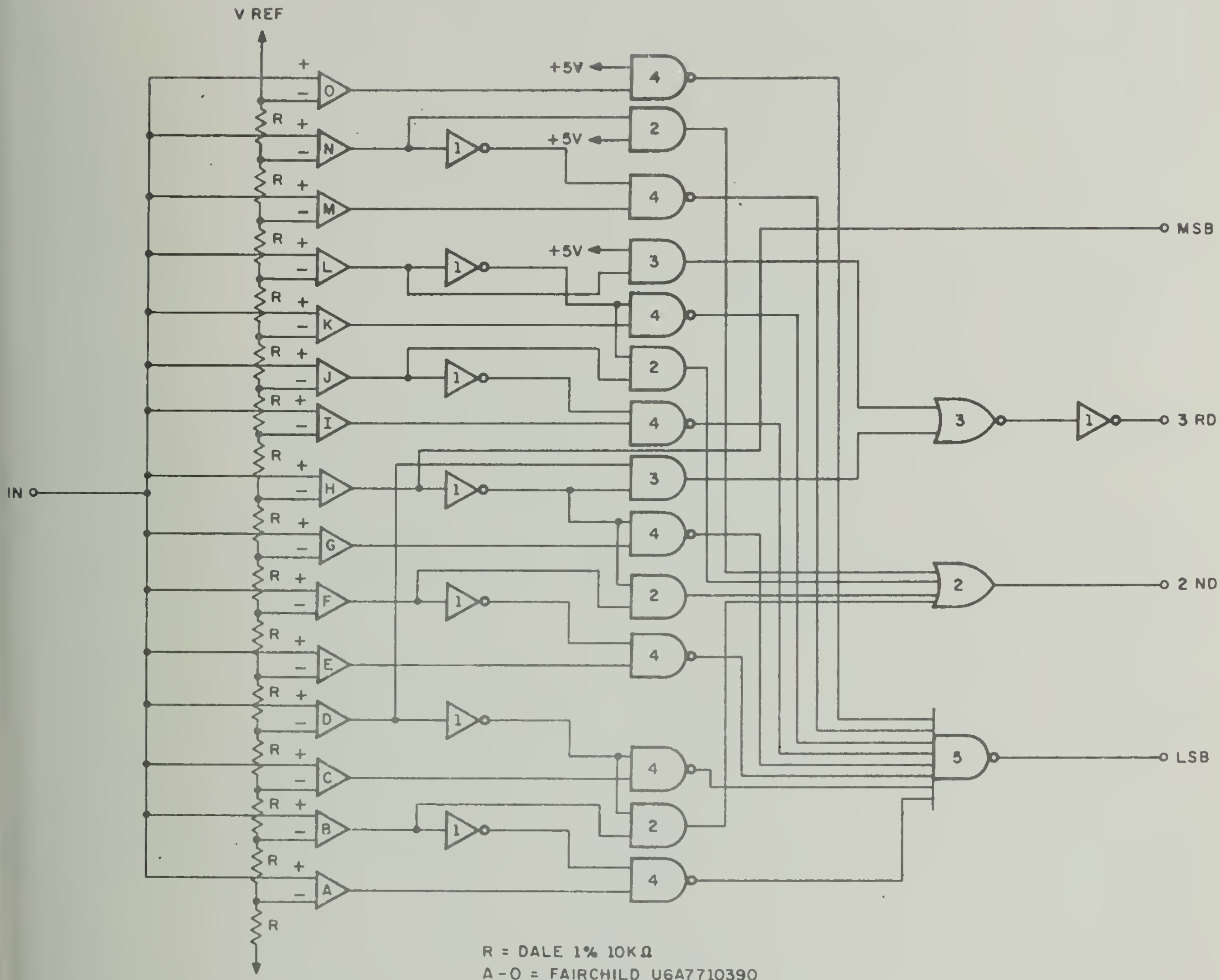
We hope to achieve three-dimensional pictures during the next quarter.

Steve Whiteside

2.7 Scantrix (Project No. 35)

2.7.1 4-Bit A/D Converter

A 4-bit analog-to-digital converter was built which would be used to change the video signal from a TV into its digital equivalent. By doing this, the digital form of the video signal can be stored in a buffer and hence "freezing" of a whole line of TV signal is achieved. Several buffers may be used to store several TV lines which are then displayed simultaneously on the corresponding LED-lines. This has the effect of improving the duty cycle of the LED's, hence allowing a brighter output picture. Improvement of duty cycle can also be achieved by mounting a storage device (e.g. a



R = DALE 1% 10KΩ
A - O = FAIRCHILD U6A7710390
1 = SN74H04
2 = SN74H52
3 = SN74H50
4 = SN74H00
5 = SN74H30

Figure 1. 4-Bit A/D Converter

capacitor) behind each LED. However, calculations showed that this latter approach is too expensive.

The A/D converter, which is shown in Figure 1, can operate up to 10 MH_z. The outputs of the ADC are given by the equations below:

MSB = H

THIRD BIT = $D\bar{H} + L$

SECOND BIT = $B\bar{D} + F\bar{H} + J\bar{L} + N$

LSB = $A\bar{B} + C\bar{D} + E\bar{F} + G\bar{H} + I\bar{J} + K\bar{L} + M\bar{N} + O$

Sik Yuen

2.8 FROG (Project No. 36)

2.8.1 Project Outline

FROG is a device which interacts adaptively with an environment. The core of the adaptive logic is a structured collection of logic units which act as a filter, modifying the feed-forward paths between input (stimul which act as a filter, modifying the feed-forward paths between input (stimul

and output. The filter is equivalent to the variable part of the world-model of the device.

All of the logic elements in FROG are drawn from a set of nine basic elements, defined hereafter, which operate in the signal continuum [0, 1]. These basic elements have been defined, functionally and mathematically, and used in computer simulations whose results have been very encouraging.

To explain the central principle of the proposed device, let us assume the context of a frog. Even more specificially consider only the feeding functions and four stimulus input (S_1 - S_4) derived from any sensory organs. At birth, the device will attempt to feed upon a relatively broad range of objects. Each feeding attempt is followed by internally-generated signals measuring the tastiness of the food object. Upon encountering a very bad tasting object, O_1 with stimulus S^*_1, S^*_2, S^*_3, S^*_4, the world-model adjusts to strongly reject as food any further object having stimulus S^*_1 and S^*_2 and S^*_3 and S^*_4, to reject less strongly any object having triple joint combination of the stimuli, to reject even less strongly any double joint combination, etc. If some other object, O_2, has bad taste and shares some properties with O_1 (say the joint occurrence of $S^*_1 \cdot S^*_3$), then its effect is to strengthen the rejection of $S^*_1 \cdot S^*_3$ as a property associated with good food, thereby tending to generalize that this joint property implies "inedibility". Figure 1 illustrates the basic concept in a 3-dimensional stimulus space. The darkness of shading is proportional to the strength of rejection of stimulus $[S^*_1, S^*_2, S^*_3]$ due to a single occurrence associated with a negative result (e.g., a bad taste). The occurrence of a negative result associated with stimulus $[S^*_1, S^{**}_2, S^*_3]$, for example will add increased rejection to the pair $S^*_1 \cdot S^*_3$ as well as to the S^*_1 and S^*_3 volumes. In other words, evidence that the stimulus $S^*_1 \cdot S^*_3$ is associated generally with "bad taste" is being accumulated. The adaptive development of the filter (partial world model) described thus far is essentially a multivalued nonlinear classification technique with continuous learning. One novelty, however, is the fact that multiple joint occurrences are adapted more quickly than single occurrences with the result that the device tends to reject large fractions of the stimulus range as food only when the evidence has become quite strong. This strategy makes possible monotone adaptation,

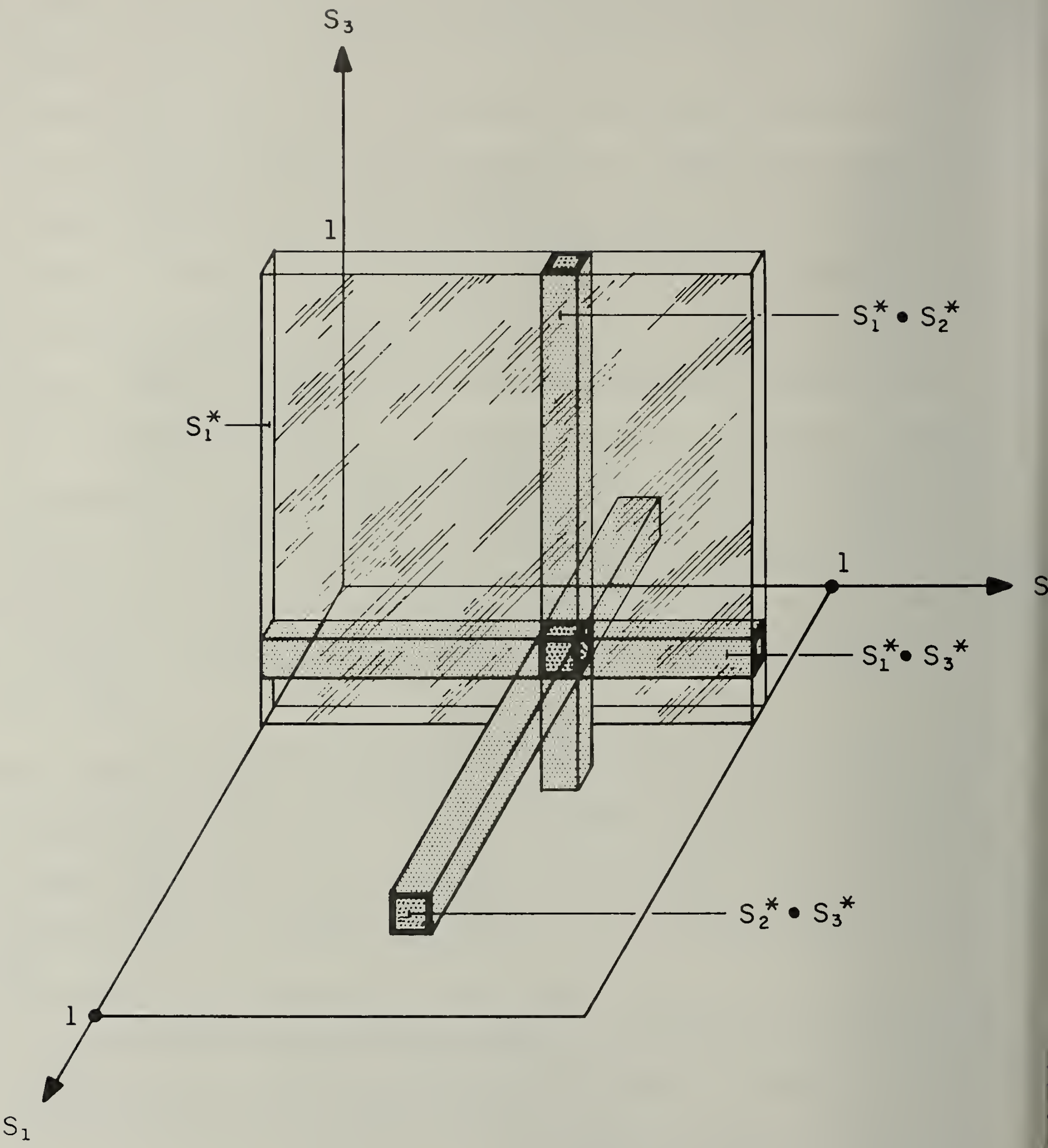


Figure 1. Adaptive Filter: Relation to a Stimulus Space of 3 Dimensions

i.e., always increasing rejection.[†] Additionally, all processes are specified in well-defined, certainly realizable, logic-like elements in a regular well-organized structure. We stress all of these qualifications because without them one obtains systems which are either very difficult to analyze or unimplementable or both.

We propose the design and construction of the autonomous mini-rooot, FROG, whose objective is to sustain its existence. FROG, tentatively a movable block, will operate on a horizontal surface having a variety of surface textures and populated by a mixture of cohabitants corresponding to potential food supply (both edible and inedible) and predators. The FROG (see Figure 2) will receive external data of two types, visual and tactile. Internally, FROG will generate hunger, fatigue, and pain signals.

The visual signals are used by FROG to locate the direction and tentative properties of cohabitants which may be potential food or predators.

Tactile data add a means of correlating the surface properties with occurrence of food or safe resting places or predators.

The central mechanism for adjusting to the idiosyncracies of its environment is the adaptive filter. A "situation analysis" is formed continuously by a mode of activity network which combines the environment sensory data, interpretated through the adaptive filter, and the somatic signals, which measure the requirements of FROG for food and rest, and decides a course of action. The three modes of activity are (1) feed; (2) rest; (3) flee.

2.8.2 Basic Elements of the Adaptive Filter

The basic elements which compose the adaptive filter are defined below. All inputs, x_i, and outputs, y, have values in the continuum [0, 1] unless otherwise noted.

[†] In simulation, we also provide a time dependent decrease of rejection, i.e., "loss of memory".

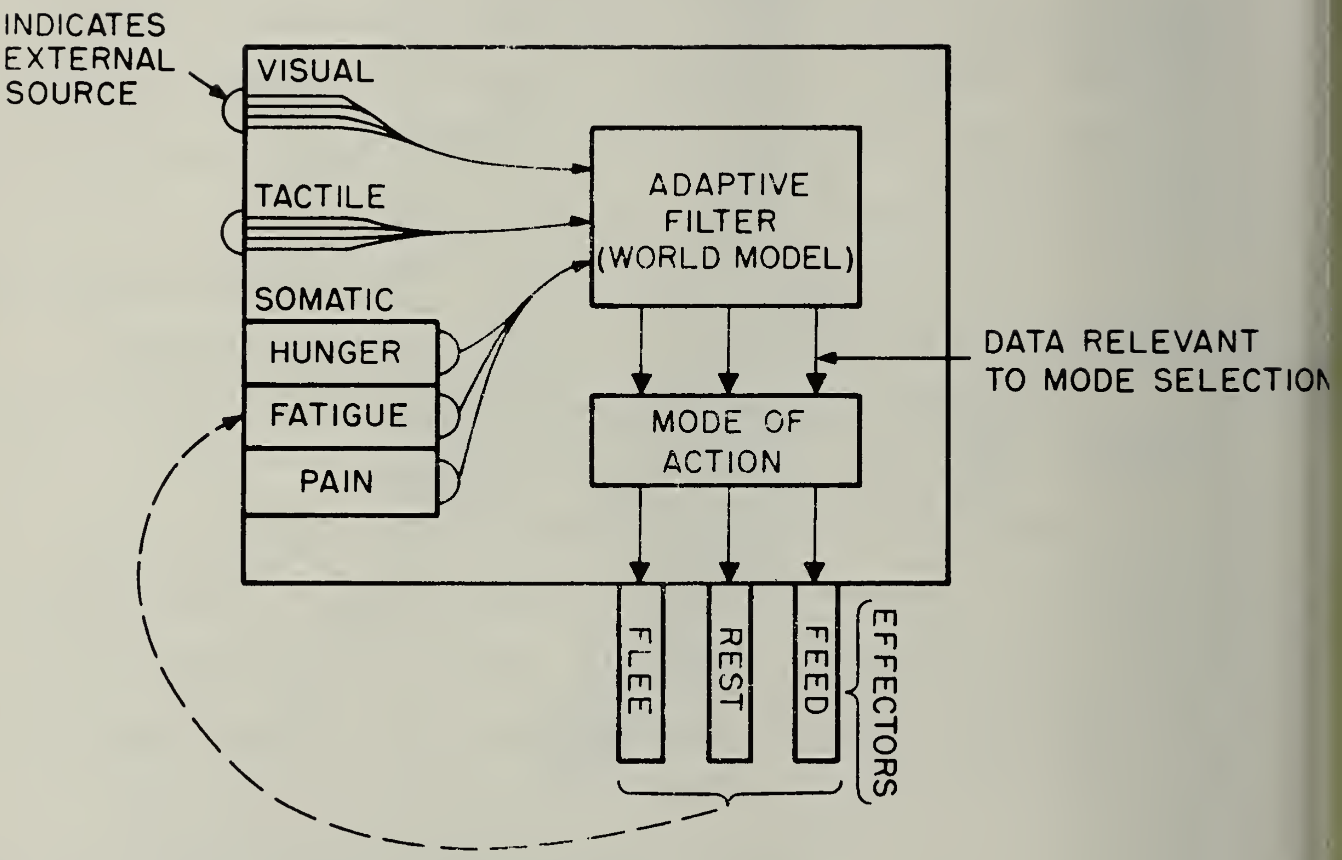

INDICATES EXTERNAL SOURCE
VISUAL
TACTILE
SOMATIC
HUNGER
FATIGUE
PAIN
ADAPTIVE FILTER (WORLD MODEL)
DATA RELEVANT TO MODE SELECTION
MODE OF ACTION
FLEE
REST
FEED
EFFECTORS

Figure 2. Functional Sketch of Autonomous Mini-Robot (FROG)

ELEMENT NAME	SYMBOL	NEAREST ANALOG	DEFINITION
Complement (C)	$x \rightarrow \textcircled{C} \rightarrow y$	NOT	$y = 1 - x$
Minimum (m)	$\begin{matrix} x_1 \searrow & \\ \vdots & m \rightarrow y \\ x_k \nearrow & \end{matrix}$	AND	$y = \min\limits_{i=1}^{k} (x_i)$
Maximum (M)	$\begin{matrix} x_1 \searrow & \\ \vdots & \textcircled{M} \rightarrow y \\ X_k \nearrow & \end{matrix}$	OR	$y = \max\limits_{i=1}^{k} (x_i)$
Product (P)	$\begin{matrix} x_1 \searrow & \\ & \textcircled{P} \rightarrow y \\ x_2 \nearrow & \end{matrix}$	--	$y = (x_1) \cdot (x_2)$
Equivalence (E)	$\begin{matrix} x_1 \searrow & \\ & \textcircled{E} \rightarrow y \\ x_2 \nearrow & \end{matrix}$	Equiv. (fuzzy)	$y = x_1$ if $[x_1 - x_2] < \epsilon$ $y = 0$ otherwise
Selective	$\begin{matrix} x_1 \searrow & \\ & \textcircled{S} \rightarrow y \\ x_2 \nearrow & \end{matrix}$	Schmitt Trigger + AND	$y = x$, if $x_2 \geq 0.5$ $y = 0$, otherwise
Sense Indicator (SI)	$x_1 \rightarrow \textcircled{SI} \rightarrow y$	Schmitt	$y = 1$ if $x_1 > \epsilon$ $y = 0$, otherwise
Memory (MEM)	$x_1 \rightarrow \textcircled{MEM} \rightarrow y$	Fed-back OR	$y = 0$ when first powered "on". If x_1' is the first occurrence of $x_1 > 0$, $y = x_1'$ thereafter
Transmittance (T)	$x_1 \rightarrow \textcircled{T}_{(n)} \rightarrow y$	(?)	$y = 0$ initially. If x_1' is the first occurrence of $x_1 > 0$ at time t_1, $y = \frac{1}{n}(x_1')$. If $i > n$, then $\frac{i}{n} \leftarrow 1$, n = positive integer.

Table 1.

The principal macroelement of the adaptive filter is a T* element (see Figure 3). A single T* element contains the relational memory between an elementary volume of stimulus space and result space. Each T* element is related to some mode of behavior by reason of its physical location in the filter.

If a stimulus arrives while RESULT = 0, then the T* element responds with an amplitude proportional to its correlation of that stimulus with the class of result. (This is a "READ" operation, essentially.) If a stimulus exists while RESULT $\neq$ 0, then a cause-effect relation is assumed to exist; the T* responds by adjusting its stored correlation value in its internal transmittance (T) element.

2.8.3 Design Progress

At present work is being done on the design of the adaptive filter, the world model. Work has been done to chalk out the details of the entire scheme. Since the objectives of the project are still in the process of change nothing is finalized yet.

Debasish Bose

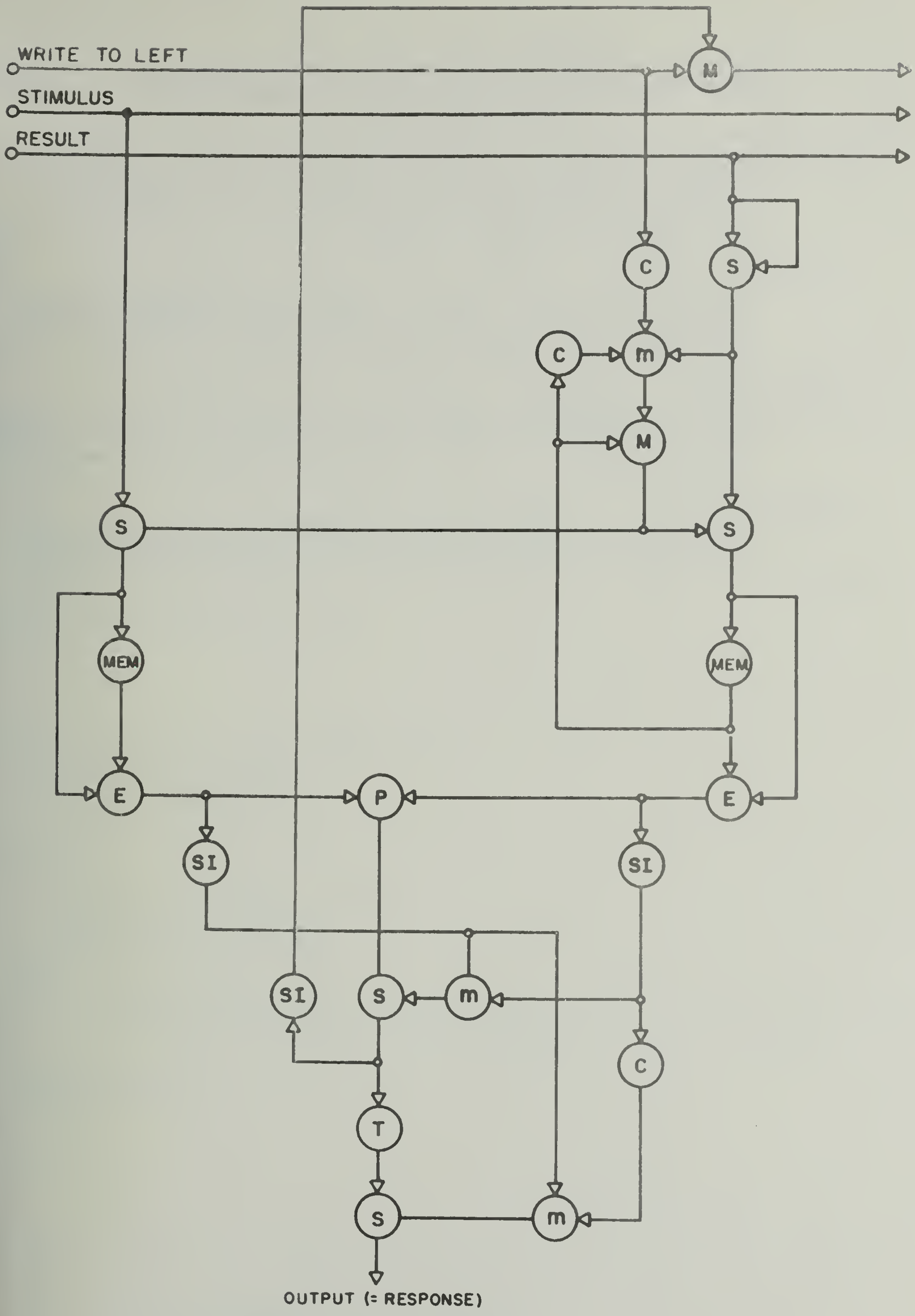


Figure 3. 'T' Element

Publications This Quarter Include:

Cheng, Richard. "Coefficient Generator and Cursor for the Stereomatrix 3-D Display System", Department of Computer Science Report No. UIUCDCS-R-71-484, used in partial fulfillment of the requirements for a Doctor of Philosophy degree.

Hadjistavros, Stavros. "COLORMATRIX. A Thermally Controlled Liquid Crystal Alphanumeric Display", Department of Computer Science Report No. UIUCDCS-R-71-486, used in partial fulfillment of the requirements for a Master of Science degree.

Kodimer, Dennis. "ALPHECON: Evaluation of a Developmental Video Storage Tube", Department of Computer Science Report No. UIUCDCS-R-71-485, used in partial fulfillment of the requirements for a Master of Science degree.

Partridge, Richard L. "PAGAN. A Three Dimensional Pattern Generator", Department of Computer Science Report No. 464, used in partial fulfillment of the requirements for a Master of Science degree.

Form AEC-427
(6/68)
AECM 3201

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIF C AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.
COO-1469-0196
Quarterly Report

2. TITLE
July, August, September

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report
□ b. Conference paper not to be published in a journal:
Title of conference
Date of conference
Exact location of conference
Sponsoring organization
□ c. Other (Specify)

4 RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.
□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

Prof. W. J. Poppelbaum
Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature
W. J. Poppelbaum

Date
September 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.
□ b. Report has been sent to responsible AEC patent group for clearance.
□ c. Patent clearance not required.

3. SOFTWARE SYSTEMS RESEARCH

A basic system for the graphical specification and numerical solution of differential equations now exists. Work this past quarter was directed towards providing a more flexible system that provides a more convenient interface for the user and works correctly with examples containing discontinuities. A complete solution to the latter problem has not been found, but an excellent start has been made.

Most of the originally planned work for the graphics end of the system has been completed for the PDP-8 system.

3.1 Numerical Processes

3.1.1 Ordinary Differential Equations (R. L. Brown)

The effects of four modifications to the original version of DIFSUB were investigated to determine if such modifications would significantly increase the rate of convergence without adversely affecting the accuracy or the amount of overhead involved in changes of step size or order. Rate of convergence was measured by the number of times the differentiating function was evaluated (number of calls to subroutine DIFFUN). Overhead calculations are primarily the number of inversions of the Jacobian matrix, PW, so the number of calls to MATINV was used as an indicator for that. The accuracy of the method was judged by comparing one of the variables Y(1,I) to the proper value of the solution function at several values of T, chosen either for convenience or because that value is a critical point in the solution. Comparison was made by evaluating the known solution F(Y,T) using double precision arithmetic and library functions, so the accuracy requested of the program (EPS = 10^{-5}, 10^{-10}) was always considerably less than the available accuracy of the solution ($\sim 10^{-15}$).

TESTS

Tests were taken from two sources: Appendix I of an as yet unpublished report by A. E. Sedgwick, University of Toronto, and "On Testing a Subroutine for the Numerical Integration of Ordinary Differential Equations," by F. T. Krogh, Jet Propulsion Lab Technical Memorandum #21F. Only tests with known closed solutions were chosen, and an attempt was made to use diverse problems affecting the numerical solution in different ways. The tests are:

1: $DY(1) = -Y_1 \qquad Y_1(0) = 1$

$\text{SOLUTION} = Y_1 = e^{-t}$

2: $DY(1) = -Y_1 \qquad Y_1(0) = 1$

$\text{SOLUTION} = Y_1 = e^{t}$

3: $DY(1) = Y_1 \ COS(T) \qquad Y_1(0) = 1$

$SOLUTION = e^{SIN(T)}$

4: $DY(1) = Y_1/4 * (1 - Y_2/20) \qquad Y_1(0) = 1$

$SOLUTION = 20/(1 + 19 * e^{-t/4})$

5: $DY(1) = (Y_1 - T)/(Y_1 + T) \qquad Y_1(0) = 4$

$SOLUTION = R = (Y_1^2 + T^2)^{1/2} = 4e^{\ /2} * e^{-THETA}$

$THETA = ARCOS \ (T/R)$

6: $DY(1) = T(1 - Y_1) + (1 - T) \ e^{-T} \qquad Y_1(0) = 1$

$SOLUTION = e^{-T^2/2} - e^{-T} + 1$

7: $DY(1) = Y(2) \qquad Y_1(0) = 0$

$DY(2) = -Y_1 \qquad Y_2(0) = 1$

$SOLUTION = Y_1 = SINT, \ Y_2 = COST$

8: $DY(1) = Y_2 \ Y_3 \qquad Y_1(0) = 0$

$DY(2) = -Y_1 \ Y_3 \qquad Y_2(0) = 1$

$DY(3) = 51 \ Y_1 \ Y_2 \quad Y_3(0) = 1$

SOLUTION = Jacobi elliptic functions

$Y_1 = 0, \ Y_2 = 1, \ Y_3 = 1$ at 0, 4K, 8K... (K = 1.86264...)

9: $R = (Y_1^2 + Y_3^2)^{1/2}$

$DY(1) = Y(2) \qquad Y_1(0) = .4$

$DY(2) = -Y_2/R^3 \qquad Y_2(0) = 0$

$DY(3) = Y_4 \qquad Y_3(0) = 0$

$DY(4) = -Y_3/R^3 \qquad Y_4(0) = 2$

$SOLUTION = Y_1 = COS(U) = 6, \ Y_3 = .8 \ SIN(U)$

U is the solution of $U - .6 \ SIN(U) = T$

$U = T$ for $T = n$, $n = 0, 1, 2,...$

MODIFICATIONS

One modification made on DIFSUB was to extend the order of the Adams-Moulton procedure (MF = 0) to 12, which required additional A(I) arrays, (I = 1, N + 2) to handle N = 12. Also, new values of PERTST were added. The scratch matrix SAVE was redimensioned to 17 x N.

Another modification involves suppressing the ability of DIFSUB to increase the step size on the step immediately following a decrease in step size. This might decrease the overhead involved in too frequent step size changes if such changes occur very often.

The number of steps the program will take after a change in step size or order before evaluating the need for another change, and the number of steps after that evaluation, if no step increase is possible, before re-evaluation, were modified in several ways. Both numbers were held at a constant value (initially at 5 and 10), both were allowed to vary as P + 2, P being the order of the method in use, and the first was kept at P + 1 while the second varied between 1 and 10.

Finally, the size of the factor that allows a change in step size was varied between 1.01 and 1.10 by steps of .01 to determine if any improvement could be made without sacrificing accuracy.

OBSERVATIONS

Of the first seven tests, only two ever made use of even the ninth order Adams method, and then only for one or two steps. Attempts to run more difficult tests have resulted in unexplained errors so no results are available for them.

The suppression of the ability to increase the step size immediately after decreasing it resulted in two more function evaluations in two tests, and identical results otherwise, possibly indicating that such occurrences are infrequent and when they happen are necessary.

It was found that the most efficient number of steps after a change in step size before evaluating further step changes was P + 1. The number of steps before the next re-evaluation, IDOVB, was varied between 1 and 10, with mixed results. See Figure 1. Anything through IDOVB = 5 gives very erratic results, although not much more can be said about IDOVB greater than 5. Since overhead supposedly decreases as IDOVB increases (this was not apparent for these tests), values of 8, 9, and 10 seem to be preferable, with 8 perhaps the best found in the study.

Finally, the size of the factor for allowing a change of step size shows erratic results for 1.01 → 1.05, more uniform performance for 1.05 - 1.08, and uniform improvement over previous values for 1.08 to 1.10. See Figure 2.

CONCLUSIONS

For stiff methods, setting IDOVB = 8 or 10 and the step size changing factor to 1.08 might result in a marginal improvement in rate of convergence without significant increase in overhead or loss of accuracy. Further study is needed on the effects of increasing the order of the Adams method to 12.

NFNS = NUMBER OF FUNCTION EVALUATIONS
NW = NUMBER OF MATRIX INVERSIONS

TEST	IDOVB = 10		IDOVB = 8		IDOVB = 9	
	NFNS	NW	NFNS	NW	NFNS	NW
2	1832	6	1342	6	1836	6
3	755	13	745	15	713	7
5	254	10	256	10	248	6
6	483	17	443	18	455	17
7	571	6	697	6	721	6
8	1260	6	1312	6	1294	6
9	2179	7	2220	7	2248	13

Figure 1

TEST	FACTOR = 1.10		FACTOR = 1.08	
	NFNS	NW	NFNS	NW
2	1312	6	1312	6
3	945	14	945	14
5	254	10	257	10
6	483	17	483	17
7	597	6	597	6
8	1260	6	1305	6
9	2179	7	2179	7

Figure 2

3.1.2 Sparse Matrix Inversion (J. S. Deogun)

The generalized eigenvalue problem may be stated as follows:

Given N x N matrices A and B, find the non-trivial solutions of

$$(A - \lambda_i B)\ X_i = 0$$

where λ_i is an eigenvalue and X_i is a corresponding eigenvector.

Our concern is the special case where A and B are of sparse nature. As A and B are assumed to be singular and of rank lower than their order, Wilkinson's* method is considered useful in this case. In fact, Wilkinson's method reduces the effective order of the matrices. In a step or two the effective order of the matrices A and B reduces to rank of B so that any standard (non-sparse) routine can be used to find the eigenvalues.

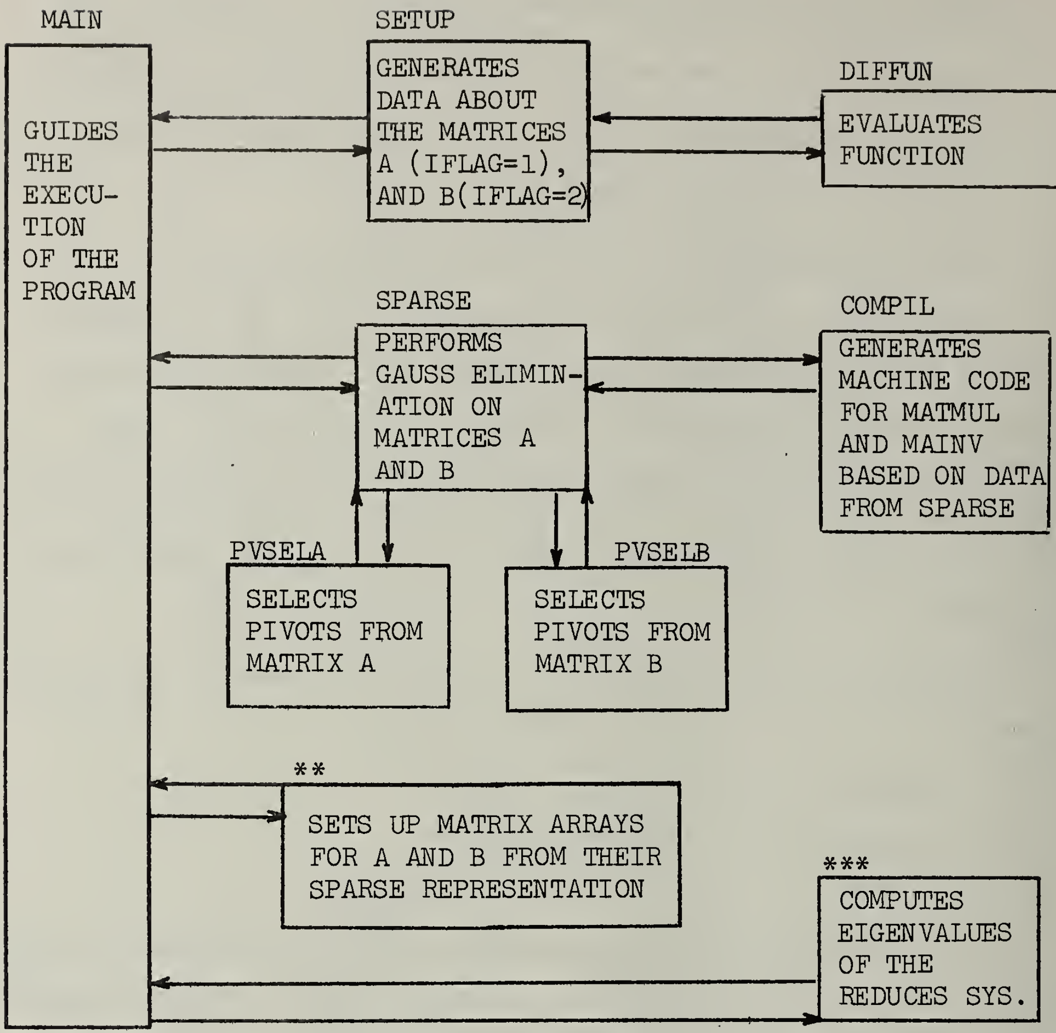
The sparse eigenvalue problem is closely related to the sparse matrix inversion and was divided into three phases:

1. Changes in the subroutine SETUP.
2. Changes in the subroutine SPARSE.
3. Changes in the pivot selection strategy.

All three phases of various changes have been computed and the combination is being run on a small test example. Relationships between various routines are shown diagrammatically in Figure 1. Following is a brief discussion of various routines and their individual functions.

Subroutine SETUP has been changed to obtain a configuration of matrices A and B which is convenient for handling by pivot selection stragegies and the SPARSE program. The configuration thus obtained allows the program to handle A and B separately as well as jointly, and as well as when required by the SPARSE program.

* For details of Wilkinson's method, see SIAM Journal on Numerical Analysis, 7, #4, December 1970.


MAIN
GUIDES THE EXECU-TION OF THE PROGRAM
SETUP
GENERATES DATA ABOUT THE MATRICES A (IFLAG=1), AND B(IFLAG=2)
DIFFUN
EVALUATES FUNCTION
SPARSE
PERFORMS GAUSS ELIMIN-ATION ON MATRICES A AND B
COMPIL
GENERATES MACHINE CODE FOR MATMUL AND MAINV BASED ON DATA FROM SPARSE
PVSELA
SELECTS PIVOTS FROM MATRIX A
PVSELB
SELECTS PIVOTS FROM MATRIX B
**
SETS UP MATRIX ARRAYS FOR A AND B FROM THEIR SPARSE REPRESENTATION
***
COMPUTES EIGENVALUES OF THE REDUCES SYS.

,* These routines will be added to the package as soon as combination of others runs successfully and all the bugs have been removed.

Figure 1

Wilkinson's method, in one step, selects pivots from B and reduces B to the form shown in Figure 2, by Gauss elimination. Exactly the same operations are performed on A. In another step pivots are selected from A to reduce A to the form shown in Figure 3. Again similar operations are to be performed on B. The SPARSE program has been modified such that when similar operations are to be performed on both matrices, the matrices A and B can be linked together in the proper way. Different pivot selection strategies are used for matrices A and B to achieve efficiency. Use of different pivot selection strategies are justified by the fact that different reduced forms are to be obtained of matrices A and B.

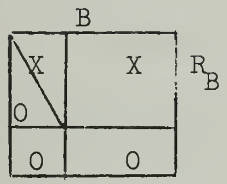


Figure 2

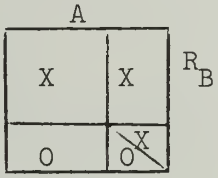


Figure 3

* R_B stands for Rank of B

As soon as the combination runs successfully on the small test example, it will be tested on big examples. This completes only the first part of the problem. The second part constitutes the developing of programs which set up matrix arrays from sparse representation of A and B and then computes eigenvalues of the reduced system by some standard procedure.

3.1.3 The Steady-State Package (B. van Melle)

The tests that have been made on the steady-state problem solver (MF=3) in DIFSUB all indicate that the best strategy for solving the steady-state problem actually has very little in common with the method used for the transient problem (MF=2). Hence a separate subroutine, DIFMF3, has been written to handle the case MF=3, while DIFSUB has been reduced to solving the single case MF=2.

The strategy used in DIFMF3 is essentially to start out with the predictor-corrector method of first order originally used in DIFSUB for this problem, but with a single corrector iteration. If no problems arise, the step size is adjusted toward one and the corrector step is phased out, resulting in simply the Newton method $y \leftarrow y - (\frac{\partial f}{\partial y})^{-1} f(y)$, which is quite fast in the vicinity of the solution. The rate at which the transition occurs is selected in an attempt to arrive at the solution quickly with a minimum number of function evaluations and matrix inversions. It is clearly necessary to re-evaluate the function at each step, but if $\frac{\partial f}{\partial y}$ does not change too fast we can avoid re-evaluating and inverting the Jacobian. So at each step $D = \sum_{i=1}^{n} |DY(i)|$ is examined to see if we are progressing towards the solution. If D has been reduced, we accept the step and continue the transition to Newton's method. However, it is not even necessary that D be reduced, and in some cases it is in fact not possible to reduce D, since we may have strayed in the wrong direction. So if D increases somewhat (the present program tolerates an increase of as much as 100-fold), we still accept the step. But if D suddenly blows up, we reduce the step size, re-evaluate the Jacobian, and repeat the step size, re-evaluate the Jacobian, and repeat the step, this time accepting it regardless. And even if no such catastrophe occurs, we periodically re-evaluate the Jacobian (presently every 5N steps, where N is number of components in vector Y), since this generally speeds convergence considerably, particularly when we are relatively near the solution.

The results of DIFMF3 tested on the various examples is shown in the following table, compared with the results from the old DIFSUB. In the simpler examples the greatest differences can be seen as a result of eliminating needless function evaluations and Jacobian inversions. On some examples the choice of initial values had a substantial effect on the performance of the old method, but the new strategy arrives at a solution with reasonable speed even where the old method had trouble or failed completely.

A new feature included in writing DIFMF3 is the ability to specify certain variables as given and others as unknowns. The present method MF=3 assumes the first derivatives $\underline{y}'$ are known and have value zero, while the variables $\underline{y}$ are unknown and are to be solved for. The new case MF=4 provides more flexibility in that the user supplies an indicator vector of the same length as $\underline{y}$ specifying for each component whether y or y' is the unknown. If y' is the unknown, the routine handles this simply by computing $\frac{\partial f}{\partial y'}$, instead of $\frac{\partial f}{\partial y}$ for the corresponding components of the Jacobian, and adjusting the value of y' instead of y in the predictor-corrector steps. The performance of this case MF=4 was tested and found to be much the same as that for MF=3. In fact, MF=3 is a special case of MF=4.

	EXAMPLE		INITIAL		OLD DIFSUB			NEW DIFMF3	
	#	N	VALUES	NS	NFNS	NW	NS	NFNS	NW
--ALGEBRAIC EQUATIONS--	1	2	1.0	134	686	122	44	58	6
			0.0	16	44	9	12	19	3
	2	2	1.0	17	49	10	15	22	3
			0.0	17	50	10	22	31	4
	3	2	1.0	26	70	9	41	54	6
			0.0	24	59	9	39	53	6
	4	3	1.0	19	65	11	9	16	2
			0.0	23	89	15	20	30	3
MIXED DIFFERENTIAL & ALGEBRAIC	5	2	1.0	18	50	10	13	20	3
			0.0	17	48	10	7	12	2
	6	9	1.0	170	1601	165	23	49	4
			0.0		FAILED*		129	213	13
	7	6	1.0	20	66	11	8	15	2
			-1.0	20	66	11	8	15	2

N - NUMBER OF EQUATIONS
NS - NUMBER OF STEPS TO SOLUTION
NFNS - NUMBER OF FUNCTION EVALUATIONS
NW - NUMBER OF MATRIX INVERSIONS
INITIAL VALUES ARE THOSE ASSIGNED TO THE VARIABLES $\underset{\sim}{y}$ BEFORE FIRST STE

* If the solution is not achieved in 200 steps we quit, but in this case the method just blew up entirely.

3.1.4 Plot Package (W. Chung)

The interactive transient analysis subroutine TRANAL (TRNØUT formerly) has been further refined and tested.

Calling sequences of P1, P2, and TRANAL are changed as follows:

1. EPS1 is deleted from the parameter lists since it is only used by P1
2. TEND is deleted from P1 and P2 lists.
3. Meanings of variables--N = # of total equations
   M2 = # of nonlinear variables

Features which make the program more flexible and sophisticated in terms of interactive performance are:

1. In P2, the user can interactively change the initial conditions for $\underline{y}$ and $\underline{yL}$ to recall DIFMF3 when KFLAG < 0 for steady-state analysis.
2. The number of integrations (INTEG) is separated from the number of points of plotting on the screen (IPNT) for the purpose of providing flexibility in plotting and reducing the time spent in numerical integration to a minimum. Numerical integration is done by calling DIFSUB as many times as the number of integrations (or maybe plottings), which is rather expensive if it is repeated every time the user wants to change the graph.

   Therefore, once a relatively large number of integrated points have been saved for a specified time limit TEND, the user can shorten or extend the time range of graph by specifying a smaller or larger IPNT ($\leq$ INTEG), respectively. In other words, the range of T on the display screen is controlled by a user such that $0 < T \leq \text{TEND}$ and

   $T = \text{TEND}*\frac{\text{IPNT}}{\text{INTEG}}$. Of course, if the user needs more graph beyond TEND he will eventually have to change TEND and the integration is re-initiated.

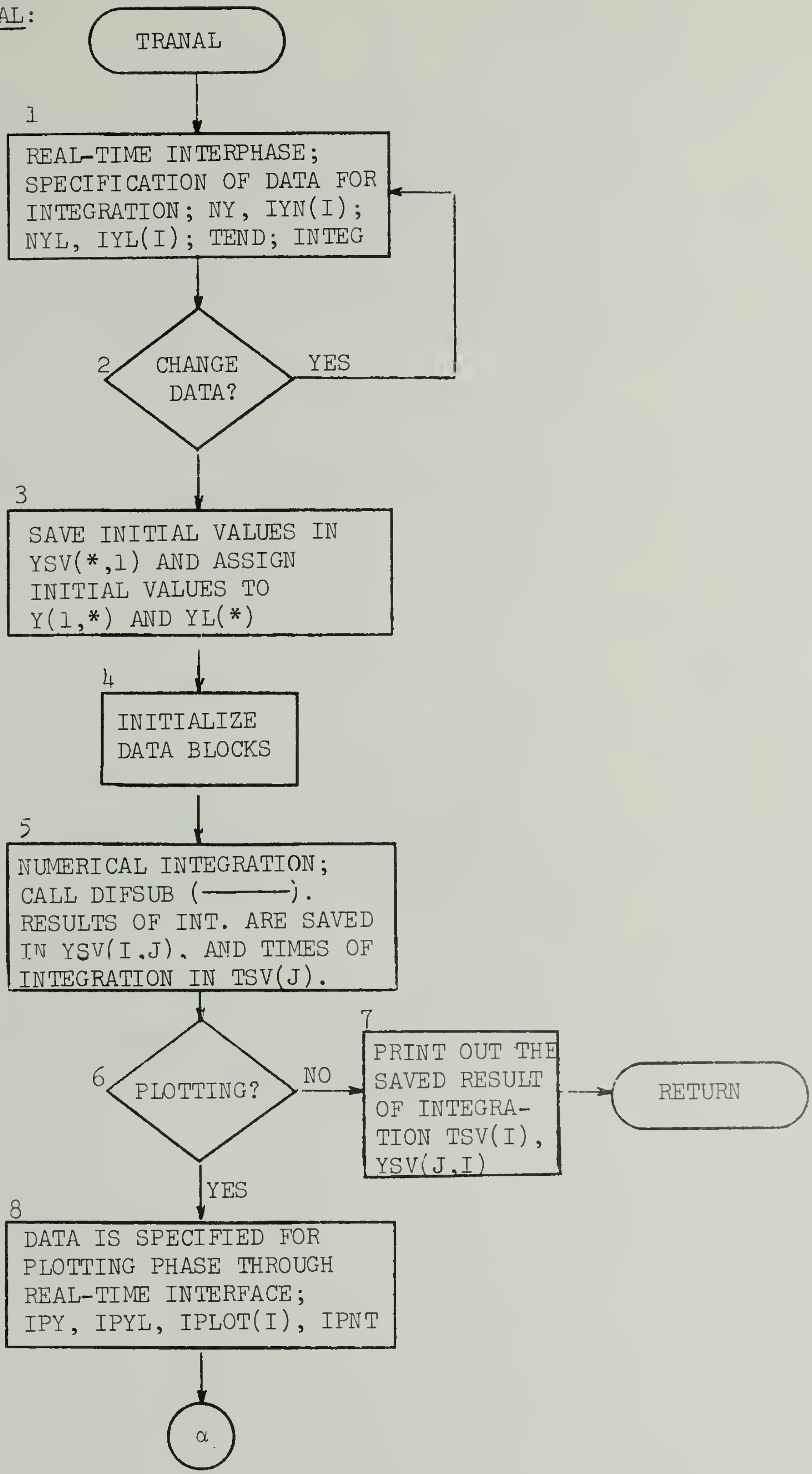
3. Coordinates on the screen can be shifted up or down to display either positive or negative values of y according to the user's request. Thereby, useless portions of the graph can be eliminated from the display screen. This magnifies the remaining portion on the screen. This is easily implemented by assigning different values to the variable IØRG which is the zero position of the T axis on the screen.

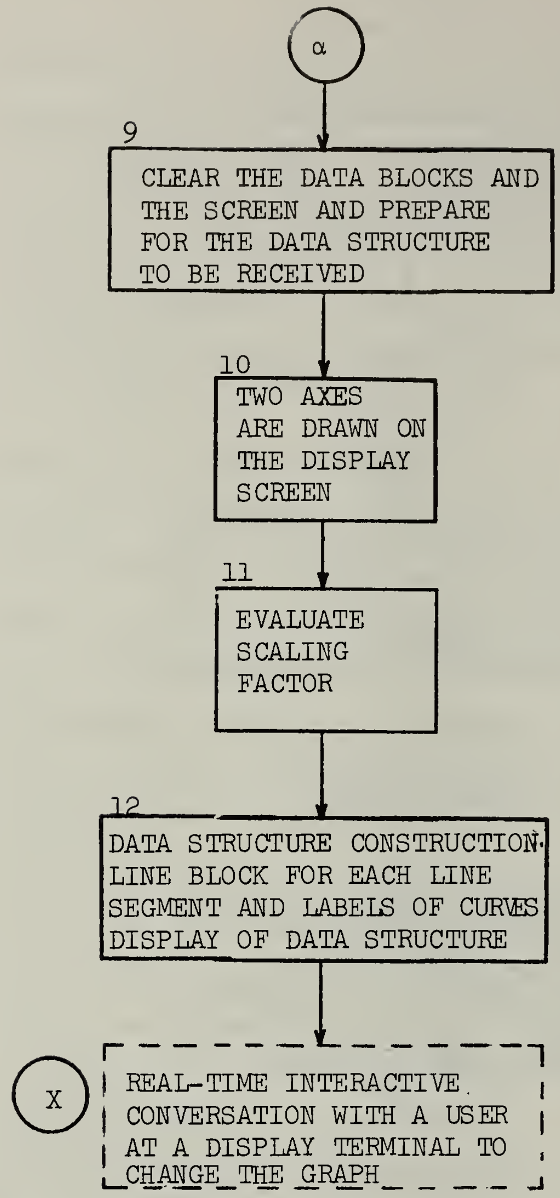
   Moving the present coordinate is not very cumbersome, as it is done by recomputing the positions of points without re-integration.

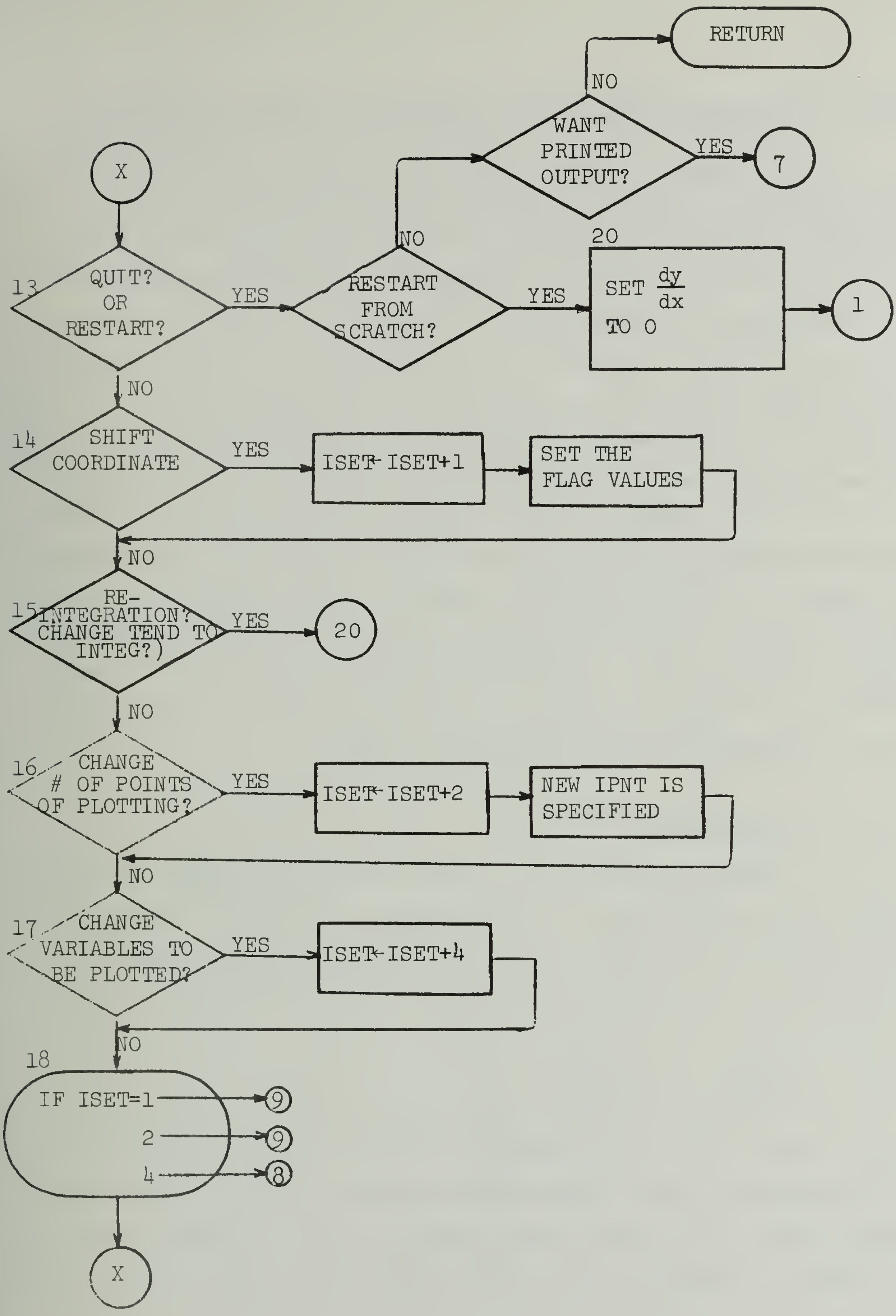
4. After completion of a graph, the user is allowed to back up to different points in the program to change parameters and regraph. Moreover, a user can restart the procedure from scratch by backing up to the initial point of the program. In that case all the derivatives of y should be set to zeros.

5. To display or receive text lines and messages--MESAGE/REPLY pair is used instead of FORTRAN READ and WRITE statements except when numerical data is included since the former is faster.

Flowchart of TRANAL:


TRANAL
1
REAL-TIME INTERPHASE; SPECIFICATION OF DATA FOR INTEGRATION; NY, IYN(I); NYL, IYL(I); TEND; INTEG
2
CHANGE DATA?
YES
3
SAVE INITIAL VALUES IN YSV(*,1) AND ASSIGN INITIAL VALUES TO Y(1,*) AND YL(*)
4
INITIALIZE DATA BLOCKS
5
NUMERICAL INTEGRATION; CALL DIFSUB (————). RESULTS OF INT. ARE SAVED IN YSV(I,J), AND TIMES OF INTEGRATION IN TSV(J).
6
PLOTTING?
NO
7
PRINT OUT THE SAVED RESULT OF INTEGRATION TSV(I), YSV(J,I)
RETURN
YES
8
DATA IS SPECIFIED FOR PLOTTING PHASE THROUGH REAL-TIME INTERFACE; IPY, IPYL, IPLOT(I), IPNT
α


α
9
CLEAR THE DATA BLOCKS AND THE SCREEN AND PREPARE FOR THE DATA STRUCTURE TO BE RECEIVED
10
TWO AXES ARE DRAWN ON THE DISPLAY SCREEN
11
EVALUATE SCALING FACTOR
12
DATA STRUCTURE CONSTRUCTION. LINE BLOCK FOR EACH LINE SEGMENT AND LABELS OF CURVES DISPLAY OF DATA STRUCTURE
X
REAL-TIME INTERACTIVE CONVERSATION WITH A USER AT A DISPLAY TERMINAL TO CHANGE THE GRAPH


RETURN
NO
WANT PRINTED OUTPUT?
YES
7
X
NO
20
13
QUIT? OR RESTART?
YES
RESTART FROM SCRATCH?
YES
SET $\frac{dy}{dx}$ TO 0
1
NO
14
SHIFT COORDINATE
YES
ISET←ISET+1
SET THE FLAG VALUES
NO
15
RE-INTEGRATION? (CHANGE TEND TO INTEG?)
YES
20
NO
16
CHANGE # OF POINTS OF PLOTTING?
YES
ISET←ISET+2
NEW IPNT IS SPECIFIED
NO
17
CHANGE VARIABLES TO BE PLOTTED?
YES
ISET←ISET+4
NO
18
IF ISET=1
2
4
9
9
8
X

3.1.5 Numerical Package (J. Frazao, B. van Melle, A. Whaley)

The numerical package was reworked this quarter. All routines besides DIFSUB were rewritten or are new. Each routine will be discussed separately. The major change was to separate portions of the matrix inversion which may be done less frequently than the inversion is now done. The techniques used identify variables and operations compiled to invert the matrix according to what they depend on. Some results of certain operations only change when the problem parameters are modified, requiring them to be evaluated only once before starting to solve a new problem. Operations of this type are now placed in subroutine MATIN1 (matrix inversion subroutine #1). Operations depending on only time and/or parameters need be re-evaluated at most once each time step, and are placed in subroutine MATIN2. Operations with results depending on other variables in the problem are placed in the remaining routine MATIN3. MATIN3 will be called when the single routine existing now (MATINV) is presently called. It is anticipated that a significant number of operations may be moved out of MATIN3.

3.1.5.1 Elimination (J. Frazao)

Changes are underway to modify elimination (WEED) to replace SETUP by doing SETUPs function symbolically. SETUP determines partial derivatives by numerical differencing, which is not as accurate or reliable, and is impossible when discontinuities are encountered.

3.1.5.2 SETUP (A. Whaley)

SETUP was rewritten to accomodate the separated subroutines outlined above. It also constructs a data structure for MATSET, telling how to take the partial derivatives numerically, where to store the results, and which variables may be differenced at the same time because no equation contains both (or all) of them. This section of the program will be retained later, when SETUP is discarded in favor of the symbolic version in WEED.

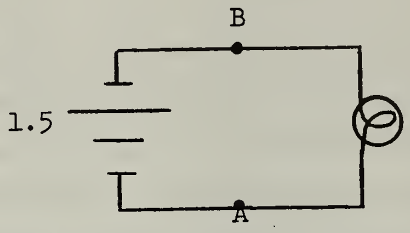
Two new routines called INSTAL and REMOVE are now used to add and delete elements from the sparse matrix data structure, making SETUP and SPARSE much easier to understand.

3.1.5.3 SPARSE (A. Whaley)

SPARSE was rewritten due to the changes mentioned above, and in order to simplify and clean up many unwieldy features of the program. An attempt was made to improve the modularization of the program by removing functions of the program to subroutines that were very messy to handle in SPARSE. Some changes were mentioned in remarks about SETUP. The decision about whether to load an operand into a register because it was to be used frequently was resolved when it was discovered that there was only one way that would give correct results (not the way formerly used). The decision about whether to reload a register because the operand is already there was removed to subroutine COMPIL.

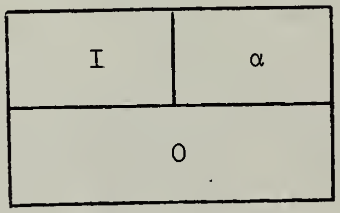
Other changes to SPARSE will result in its being able to handle more different kinds of problems. As SPARSE inverts a matrix, the matrix cannot be singular, which it almost always is. Ways were, therefore, investigated to handle singular matrices. It has been found to be common (and correct) to occasionally have an extra, redundant equation in the system of equations, leading to linearly dependent rows in the Jacobian coefficient matrix. The nonsquareness of the matrix is, therefore, ignored, and as long as the numerical integrator can solve the equations, only a nonfatal indicator message is given to the user. It is usually not possible to identify the useless equation, especially since it is generally possible to remove any of several. For this reason, the system does not attempt to identify extra or missing equations. An independent program by John Koch will be constructed to give the user a map of his equation structure so that he may easily identify the missing equation.

In the case of missing equations, after reducing the matrix to its most basic form it is discovered that the values of some variables must be specified (in a sense supplying the missing equations) before the integration may be continued. These variables are set to zero, and the user informed. Oddly, this is a normal occurrence, as in the following example:



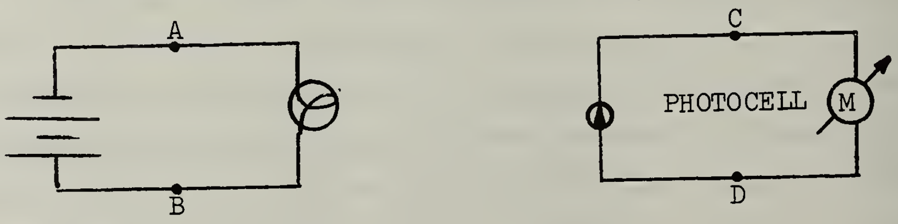
It is specified that the difference in voltage V(B)-V(A) is 1.5 volts, but what is the voltage at A? One of these variables (A or B) will be selected and set to zero. The user really is not interested in this process at all.

A reduced matrix appears thus:



where I is an identity matrix, 0 is zero, and α is a reduced set of equations that cannot be solved. Each variable in α can have its value specified.

The following problem was considered:



In this problem either V(A) or V(B) should be specified (also V(C) or V(D)). At first it was wondered whether the section α would contain one variable from the set of A and B,and one variable from C and D or an incorrect combination such as C and D with none of A and B. This speculation arose because of many other problems caused by not knowing the values of the elements in the matrix while blindly generating code to invert it. It appears (not proven, however) that α will always consist of a correct combination. Different pivot strategies will affect whether A or B is one of the unspecified voltages, but it does not appear possible for both of them to appear in α as unspecified.

Inconsistent equations are detected by the numerical integration package when it is discovered that a given equation cannot be satisfied. It is then known that this equation or one of the other equations, multiples of which were added to this one during the Gaussian elimination, must contain the error. Identifying this subset of equations when trouble occurs has not yet been implemented.

In general, the requirements that the Jacobian be square or non-singular have been suspended.

3.1.5.4 MATSET (A. Whaley)

This program has been modified so that it only computes partials for a given subroutine number (1 to 3 for MATIN1, MATIN2, and MATIN3). In addition, MATSET calculates the partials for several variables simultaneously in all equations. The group of variables involved do not appear in any equation together (see the description in SETUP). In other respects this program resembles its predecessors.

3.1.5.5 COMPIL (A. Whaley)

This program was virtually coded in absolute machine language, so no attempt was made to work with the old copy. A new version was written with several improvements besides handling multiple subroutines. Each subroutine starts off with 1024 bytes, a good size to avoid fragmentation with IBM's defective GETMAIN routine. The old COMPIL had a fixed area of 8K each for MATINV and MATMUL. The four subroutines now being compiled each start at 1K, but may be extended by 1K increments (separately) to any size.

3.2 Non-Numerical Packages

3.2.1 Item Analysis (J. Koch)

Item analysis worked correctly for all networks put through it this quarter. Work was begun near the end of the quarter on generating an extra equation and variable for each occurance of the natural logarithm, and square root functions. These extra variables would then be used to avoid the points of singularity and complex results.* Two other routines were written, FSCAN and SPIEXT. Called from a FORTRAN program, FSCAN will scan a buffer for a blank, equal sign, or comma and return the previous characters in another buffer. Also, if the scanned characters were legal numbers, they were converted into a floating point number. The calling sequence is as follows:

```
CALL  FSCAN(INBUF,INLEN,OUTBUF,OUTLEN,STACHR,ENDCHR,CONVER,
      STPCHR,LENMOV,CODE)
```

where:

INBUF - the input buffer where the characters to be scanned are located.

INLEN- length (in bytes or characters) of the input buffer.

OUTBUF - the output buffer where the characters scanned from STACHR to the delimiter are placed.

OUTLEN - length of the output buffer.

STACHR - beginning with 1, the character position of the first character to be scanned. This parameter is updated to one character past the delimiter upon return to the mainline.

ENDCHR - last character to be scanned.

* These functions have caused problems in the search for an initial steady state. Since "random" initial values are inserted to develop the sparse matrix structure and to start the iteration for initial values, the appearance of SQRT(exp) in another expression can lead to a negative value for exp. This is overcome by adding the variable Z, which is constrained to be initially positive, replacing SQRT(exp) by SQRT(Z) and adding the equation Z - exp = 0 to the system. The Newton or other iteration will then restrict the changes to Z such that it never goes negative.

CONVER - word where the floating point number corresponding to the number scanned is placed upon return (see CODE).

STPCHR - the delimiter (blank, comma, or equal sign) will be placed in the first byte of this word upon return.

LENMOV - an integer count of the number of characters moved into OUTBUF.

CODE - '0' = the characters scanned were not numbers.

'1' = the characters were numbers and the floating point representation is in CONVER.

The routines CHRBIN, CHRFLT, FROUND, and TABLE1 are used to do the conversion from the EBCDIC to binary floating point.

The routine SPIEXT is used to patch up any floating point overflow, underflow, or division by zero detected by the 360 operating system. Without this routine, the equation analysis would be halted whenever one of these errors detected, and the numerical package would not be able to correct these errors by other means. In order that the operating system know where to go for overflow, underflow, and division by zero errors, a SPIE macro must have been previously executed with entry point 'SPIEHOLE'. When one of these errors occurs, the result register is replaced by X'3F8 00000 00000000' or D'0' (for underflow), an error count is incremented by 1, a search is made to pinpoint the beginning of the erring equation, and execution is allowed to continue. After an equation or set of equations has been run, a function call to SPIEXT is made (Note: the call must be 'SPIEXT(Ø)'; Ø is a dummy variable that is not used or modified. This is because FORTRAN expects at least one argument in each function call). The error count is returned in register 0 and the error count is zeroed out.

3.3 Graphical Remote Access Support System (GRASS)

Considerable effort this quarter has gone towards documentation and consolidation of the system. Further debugging also took place, with the result that the local package is now quite stable. Several important additions to the system were made and are documented below.

3.3.1 Disk Monitor System (R. Haskin)

Reasonably reliable production use of the DMS continued this quarter. The disk was returned to Data Disk for repairs on August 16, and has not come back up since it returned. In the interim, a DECtape system program called "DMS" was developed to allow the use of the disk monitor from Swaptapes. "DMS" consists of a new monitor head which maps disk I/O requests (SYSIO calls) to a Swaptape on unit 8. When run (by typing "DMS" instead of "SWAP" at the start of a programming session), the replacement monitor head is read-in in place of the standard one, and operation can proceed in a normal, if grossly slow, fashion.

3.3.2 Display Terminals (R. Haskin)

ACID continued to perform well this quarter. Continued hardware debugging plus some software changes contributed to a decrease in "garbaged" pictures. ACID modifications were

1. Changing all IOT wait loops to time out and go to an error routine if the associated device is not ready. This corrected problems with the terminal not returning coordinates after read-cursor commands, and with users inadvertantly turning off the terminals or the PDP-8.

2. Modifying the display interrupt handler to be more intelligent about releasing buffers when done displaying segments, thereby improving memory utiliztion.

3.3.3 Information Retrieval (R. Haskin)

The package sent to Griffiss Air Force Base has been successfully integrated into their system.

The local version was in production use with GRASS all quarter. Two modifications are in progress, the first implementing user codes in order to allow some form of file separation and protection; the second being the investigation of the final implementation of a more sophisticated protection scheme.

3.3.4 Monitors (M. J. Michel, R. Haskin, J. Nickolls)

Several reports describing in detail the operation of the system were written this quarter, including:

SYSTEM OVERVIEW, Report 465
REMOTE FACILITIES GUIDE, Report 466
TERMINAL USER'S GUIDE, Report 467
SYSTEM SOFTWARE DESCRIPTION, Report 468
EXTENDED REMOTE FACILITIES GUIDE, File 867

GRASP (360 Remote Monitor)

Production use has been reliable all quarter, no changes were made or are anticipated.

2701 Data Link

The interface between the PDP-8 and the 2701 PDA has been reliable all quarter.

GLASP (PDP-8 Local Monitor)

Production use has continued reliably all quarter.

Program Segments

CALLER: The "purgeout" feature has been changed from a menu pick function to a command which must be typed in on the keyboard. It was felt that this was necessary to prevent users from accidentally doing a purgeout by inadvertently hitting the wrong screen area.

GNDRW: Several important functions have been implemented this quarter.

All move and delete functions are currently implemented, allowing easy modification and correction of an existing picture.

The "display all terminal connections" function has been added to allow the user to see the invisible as well as the visible terminal connections.

A "picture recompile" feature has been added. This function, invoked from the keyboard via a R&<mnemonic name>, "recompiles" the picture to include the latest instance definition of the specified mnemonic. The instance and terminal blocks are modified to reflect any changes which may have been made to the instance definition since the picture was constructed; and error checking is done to insure that the new definition is at least topologically equivalent to the old one. The feature eliminates the need for deletion, respecification and reconnection of each instance of a mnemonic when its definition is changed, making it a great pain and work saver.

Scheduled for implementation in the next quarter are all currently unimplemented functions plus a feature allowing horizontal and vertical constraints to be applied to all drawings to enable construction of neater looking pictures.

New Program

LLSD: Local Library Service Discographer

A new program segment was written this quarter to facilitate maintenance of the local PDP-8 library.

LLSD is selected from the menu in CALLER.

The user's menu is displayed and may be scrolled up and down by joysticking the PREVIOUS and NEXT light buttons as in GNDRW.

All commands to LLSD are entered via the keybaord. Only the first two characters of the command are examined, and a space separates the command from the first operand. Two commands are currently implemented: COPY and REACT.

COPY copies the first file to the second file. A space separates the two filenames. A file may be a mnemonic instance or a picture. Filenames must contain a picture or mnemonic name, optionally followed by a user ID. If no user ID is specified, the user's logon ID is used. Filename conventions are identical to those used in LSD (see DCS Report No. 466), with the omission of the library name.

Some examples follow:

```
COPY_PICTUR/ØØ72POOHBEAR_MYPICT
COPY_DIODE.1_DIODE.1/9999CSJOCK
COPY_NEURON.6_NEURON.4
COPY_NET5/ØØ72NICKOLLS_NETWRK/ØØ72HASKIN
```

REACT has no operands; it transfers control to the program segment REACT for access to the remote 360 library via LSD. Joysticking the RETURN box in REACT will return the user to LLSD. This feature allows interaction between LSD and LLSD, facilitating communication between the remote library and the local library.

Several additional functions for LLSD are being planned: a PROTECT command for specifying protection for any file, a CATALOG command for displaying a user's menu, and a DISPLAY command for displaying a user's file.

GLASP Status

The monitor and all program segments appear at this time to be quite stable. With the publication of the five reports mentioned earlier, it is also felt that the system is well enough documented to allow new users to familiarize themselves with the system and use it with a minimum of trouble. Software documentation is also in a state where it would not be difficult for a competent programmer to use, add to, or modify the existing system software.

Applications Support

The extensive use of the package for the development of the Simulation and Modeling System has continued this quarter. Many networks have been specified and sent to the remote system for analysis. Work is currently proceeding towards implementation of graphical communication in the opposite direction.

A generalized graphical output program is currently being written to provide visual output from the integration routine through the motions of objects on the screen.

A user at a terminal may visually observe the progress of the integration phase in the solution of his problem. The user first creates pictures to be moved later across the terminal screen. He then associates the coordinates (size and rotation optional) of each picture with a pair of variables being integrated in the problem; one variable for X and one for Y. A new picture is sent to the terminal at each step in the integration, causing the pictures to move as the values change. A "trailer" may be attached to a moving picture to record its path, or successive pictures can be displayed over each other without erasing in between. For example, if the picture is the letter "O", a "trailer" would produce a conventional graph with circles at the data points and straight lines connecting the points.

More elaborate pictures could display movement of bodies in potential fields, traffic density and flow in a model intersection, or deflection in a structure due to stress and strain.

A program using graphical input and output for investigating methods for the numerical integration of stiff equations has been developed using the remote data structure utilities (COMMUNE, etc.).

Illustrating the generalized capabilities of the GRASS system, work has begun on the implementation of a graphical matrices language based on the OL-2 language and using the terminals for graphical input and output.

3.4 Computer Maintenance and Construction

3.4.1 Graphics-8 Hardware (C. E. Carter)

The total Graphics-8 hardware has been moved next door to room 31J and up on a false floor. The operation, maintenance, and demonstration of the equipment are much better than before. Also, there should be some improvement in reliability of the system.

All the time the hardware was powered down for the move, the Data Disk was shipped back for examination and overhaul at the California plant. This disk has been in use for some time during the developmental phase of the project and will have to be applied during the time that a search for a replacement is made.

New Disk Acquisition

While this was proceeding, a bid specification was prepared and sent to possible suppliers. The "hard" part of the specification and the analysis sheet is contained here.

Specification of Rotating Magnetic Storage Device

"The device in question is a rotating magnetic storage device (disk or drum) along with its controller. Its interface requirements are those established by Digital Equipment Corporation in the (DM01) multiplexor on the PDP-8 processor. The performance specifications are:

1. Capacity (Min. useful data) 256K (12 bit) words,
2. Block Size Variable from 1 word to 4096 words,
3. Average Latency (No greater than) ~ 17 milliseconds,
4. Transfer time/word (No greater than) ~ 19 milliseconds,
5. Reliability--No more than one nonrecoverable error in 10^8 bit transfers.
6. Interface requirements--PDP-8 levels of 0V. to -3V. or TTL levels of 0V. to +3V. (Preferably not mixed).

In addition to the above, detection and correction of recoverable and detection of nonrecoverable errors shall be provided. All of the testing and checking required to establish reliability or to carry out diagnostics is assumed to be a part of the disk, its controller or the software supplied with the disk or its controller."

BID ANALYSIS (K000017) October 1, 1971

COMPANY	PRICE $	DELIVERY	WARRANTY	TRAINING	EXPANSION	COMPUTER SOFTWARE	AV. ACC	WORD XFER	SECTORS/ TRACK	WORDS/ SECTOR
1. Jackson Assoc.										
Xerox Disc	16K									
2. DEC	15.58K									
3. Digital Develop. Corp.		90 days	1 year							
Single Disc			90 days	$900						
Single Cont.	10.635K		Cost Free							
Single (Disc-Exp.) Cont.	10.935K				Double Heads	Not Mentioned	8.8ms	235KHZ	256	16
Single Disk Dual Cont.	11.935K				Additional Drives					
4. System Industries AMC Disc	12.22K	60 days	1 year all labor and parts	No Charge	Can double heads--can add 3 additional drives	Can use DEC software	8.7ms	255KHZ	16 or 32	256 or
5. Data Disc	11.2K	60 days	2 years 90 days Cost Free	$1200		Patches Provided	16.7 ms	125KHZ	64	128

The work resulted in our requesting a disk and controller from System Industries.

3.4.2 Equipment Maintenance Log Summary (H. Lopeman, B. Miller)

PDP-8

1. SW register noise on PDP-8 console. Wiggling switches will cause the processor to halt. (No solution as yet)

Inktronic

1. Paper feed clutch frozen. (Lubricated and cleaned)
2. Blown high voltage power supply. (Replaced supply)

PDP-8/I

1. High speed reader motor bad. (Replacement motor ordered)

Computek

1. Terminal #2, bit 2 on board "B" intermittent. (Cleaned and repaired board, OK)
2. Terminal multiplexor still giving spurious problems.

PDP-7

1. Several problems existed in the PDP-7 during this period.
    a. The instruction decoder area of the 2701 interface contained two bad nand gate boards.
    b. The interrupt area of the PDP-7 CPU contained a bad pulse amplifier.

c. The program counter of the PDP-7 CPU had a bad flip flop board, and a carry chain pulse amp.

d. The paper tape punch was cleaned, lubricated, and adjusted.

Boards replaced in the PDP-7 were

1L26-B204
1D23-B201
1D29-B201
1E24-B620

All filters have been changed and cleaned in the PDP-7, 630, PDP-8, and 338 during the month of August 1971

3.4.3 Stereomatrix Interface (I. Cunningham)

The interface logic for the stereomatrix display and the PDP-8/I has been checked out and is now working. The display was simulated by a square wave generator in order to check all functions. A pushbutton box has been constructed. The logic control for it at the display end is now being wired.

A software system has been organized and memory allocators, display and clock interrupt controls, basic file management, and straight line generator routines have been written. This basic system will permit the drawing of simple 3D objects. The organization and coding of these routines will hopefully aid in identifying system pathologies.

All hardware work is analyzed until the stereomatrix display comes up. Checking and expansion of the software will continue in the next quarter.

Reports Printed

Haskin, R., Nickolls, J. and Michel, M. GRASS: Remote Facilities Guide, Department of Computer Science Report No. 466, University of Illinois, Urbana-Champaign, July 1971.

Michel, M. GRASS: System Overview, Department of Computer Science Report No. 465, University of Illinois, Urbana-Champaign, July 1971.

Michel, M. and Koch, J. GRASS: Terminal User's Guide, Department of Computer Science Report No. 467, University of Illinois, Urbana-Champaign, August 1971.

Michel, M. GRASS: System Software Description, Department of Computer Science Report No. 468, University of Illinois, Urbana-Champaign, August 1971.

File Numbers Printed

Haskin, R. GRAPHICS 8: System Maintenance (Software), Department of Computer Science File No. 865, University of Illinois, Urbana-Champaign, July 1971.

Michel, M. and Haskin, R. GRASS: Extended Remote Facilities Guide, Department of Computer Science File No. 867, University of Illinois, Urbana-Champaign, August 1971.

Nickolls, J. and Michel, M. GRAPHICS 8: System Maintenance (Hardware), Department of Computer Science File No. 866, University of Illinois, Urbana-Champaign, August 1971.

Form AEC-427
(6/68)
AECM 3201

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY–TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1 AEC REPORT NO. COO-1469-0196	2. TITLE 3rd Quarterly Progress Report 1971 (July, Aug., Sept.)

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report
□ b. Conference paper not to be published in a journal:
Title of conference ____
Date of conference ____
Exact location of conference ____
Sponsoring organization ____
□ c. Other (Specify) ____

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.
□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)

C. W. Gear, Professor
and Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature Charles W Gear.	Date October 27, 1971

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.
□ b. Report has been sent to responsible AEC patent group for clearance.
□ c. Patent clearance not required.

4. IMAGE PROCESSING AND PATTERN RECOGNITION RESEARCH: ILLIAC III (Supported in part by Contract AT(11-1)-2118 with the U.S. Atomic Energy Commission)

A balanced program of experiments with an image processing computer is being carried out. Developments of this past quarter include:

1. Atlas, an extension of Show-and-Tell, our package for interactive picture processing, is being designed and implemented to provide the capability of correlating graphically-defined Atlas information with corresponding pictorial information. (See Section 4.1.2)

2. The flying spot microscope became operational during this quarter. Full X, Y, and Z axis motion is possible and under PDP-8e program control. (See Section 4.1.4)

3. Development of the theoretical basis (signal detection theory, covering techniques, etc.) for plane parallel image processing procedures critical for rapid preprocessing and local feature extraction is being developed. A report entitled "Texture Analysis" by S. N. Jayaramamurthy is in final editing. (See Section 4.2.1).

4. Cervical smears: Using the PAX II package running under Show-and-Tell on the IBM 360/75 and PDP8/e, programs have been written which can perform attention-centering and some feature extraction on single, selected microscope fields where the degree of cell overlapping, clumping and degeneration is not too high. (See Section 4.4.1)

5. Transfer Memory II: The design phase is now largely complete and construction bids are being prepared. (See Section 4.5.3)

6. Taxicrinic Processors: Work over the past quarter involved the publication of Volumes Two and Three (DCS Reports #475, 476) of the Taxicrinic Processor (TP) design. The design of this basic machine is essentially complete.

4.1 INTERACTIVE PICTURE PROCESSING

4.1.1 Show-and-Tell

During this quarter, Show-and-Tell (S & T) was used extensively as a research tool, primarily in the development of programs for analysis of cervical smears (see Section 4.4.2). At present all image processing is done on the IBM 360/75 of the Computing Services Office, with the local S & T software performing image acquisition and display and picture transmission back and forth between the 360 and the PDP8/e.

Application development has now progressed to the point where it is necessary to be able to process large quantities of pictorial data. This is not feasible to do on the 360/75 for economic, technical and administrative reasons. Also, the hardware Pattern Articulation Unit is nearing the checkout stage. Because of these considerations, we are implementing the local image processing functions of S & T, as defined in the S & T System Specification (DCS Report #429).

This quarter, some essential subroutines for a software PAU simulator were completed. The simulator will be used for PAU hardware checkout and debugging, as well as for image processing. As the hardware PAU becomes available, the simulator modules can be phased out.

4.1.2 Atlas Extension of Show-and-Tell

Atlas, an extension of Show-and-Tell, is being designed and implemented to provide the capability of correlating graphically-defined Atlas information with corresponding pictorial information.

Areas of application include the following:

1. Brain mapping at a gross anatomical level.
2. Correlating satellite multispectral data with topographical

and geophysical maps.

3. Fault diagnosis of LSI circuitry, given a scanning electron microscope as a digitally-directed probe and layouts/diagrams of the LSI circuitry.

4. Interpretation of microscopic pathology in histological sections, given an atlas of descriptive histology.

5. Correlating anatomical information with X-rays, e.g. from textbook illustrations of the spine interpret an X-ray of the same.

6. Correlating remotely-sensed thermal and radiation pollution of urban areas with area maps.

Ideal Capabilities of the System

1. The Atlas material - considered here as largely comprising a set of line drawings - would be read into the machine through any SMV input port: TV camera or flying spot scanner (using a film intermediary). This graphical material is to be digitally encoded and stored.

2. The image material also would enter the system through any SMV scanner or TV camera. This material normally would not be digitized or only selectively digitized in a manner described below. Immediate requirement is that the graphically (i.e. Atlas-) derived information can overlay the optical image and be used by the operator to interrogate the incoming scanner image.

3. In general the optical image will exhibit distortions (of stereo-projection, normal biological variation, etc.) and will also include areas which will match incorrectly or incompletely the Atlas information. Normally, it will be necessary to correct or distort the Atlas-derived graphical image to enhance its match to the optical image. This capability

requires two largely separable facilities:

i) The ability, using the available Atlas information, to detect the actual boundaries in the picture and therefore parameterize the required deformations and corrections of the Atlas-derived information.

ii) The ability to generate the correct deformation or corrections of the graphical image.

A rather simple case of the above is the facility to scale the Atlas materials to match rectified images of the corresponding areas.

4. The Atlas information, though entered into the computer possibly as a set of serial cross-section graphs, would be handled internally, when applicable, as a coherent three-dimensional model. For example, in brain mapping we wish to speak of the boundary of a given nucleus. Also cross sections (sagittal, coronary, etc.) perpendicular to more than one axis may be required.

5. There is also the necessity of realizing that the image data and possibly the Atlas data may be viewed at more than one resolution. Specifically, with a slide under the microscope, we would like to zoom up and down in resolution by changing the microscope objective and to systemize this additional information in an augmented Atlas. Changes of magnification will be needed, e.g. in brain mapping to identify nuclei boundaries.

6. The graphical information will be coordinated by a hierarchical structure of directed labeled graphs. For manipulating these latter, it is hypothesized that a variant of the Structure Operation Languag may be applied. Pointing to a region on the CRT can be condsidered as generating a pointer to the region node in the appropriate digraph structure.

7. The Atlas information, particularly the digraph structure, should also play a key role in later extensions of the system to provide information storage and retrieval. Specifically, pharmocological reports (journals, articles, etc.) noting drug responses can be indexed by the anatomical site of excitation/detection. Analogous techniques can be devised for other Atlas-described materials.

8. Journal illustrations, particularly in electrophysiology but also in agriculture and geophysics, can often be considered as providing augmented local Atlas information. Accordingly, the graph structure representing the Atlas should be considered to be growing and constantly modified as new scientific reports appear.

9. Large-scale files (see McCormick and Richardson, "Design Concepts for an Information Resource Center with Option of an Attached Automated Laboratory", DCL Report #203) are now becoming commercially available. In particular, Foto-Mem, Inc. is now building for the New York Times a retrieval system where the information retrieved is stored in microimage form. In general, the card handling capabilities built into the trillion-bit photo memories using laser-generated bit patterns can be and are being commercially adapted to provide large-scale microimage storage. It is anticipated the Atlas system could evolve to provide a powerful storage and retrieval mechanism for such image-based systems.

Implementation

1. The system cannot be implemented at a single pass since extensive experimentation is needed at each step. Accordingly, thus far the implementation has proceeded as a series of small experiments. The experiments have been ordered so as to provide us enhanced image processing

capabilities at each stage.

2. Implementation studies decompose into two largely separable activities:

i) Initial encoding, analysis and representation of Atlas information. Here the existing Show-and-Tell System is being used to digitize Atlas "cartoons" and associated image data. The graphical overlay procedure makes extensive use of the incremental mode of display at the monitor.

ii) Analysis and implementation of the Graph Structure Language (see Section 4.3.2) and its associated file system.

3. The use of the Atlas-derived information to control image processing is being left largely to individual investigators (e.g. brain mapping. See Section 4.4.2). Many general tools are becoming available here. These tools include: Interval Covering Techniques, Binary Factorization Procedures, Clustering Techniques, and the formal thesis work of S. N. Jayaramamurthy and J. C. Schwebel.

Progress on Atlas Transformations

Since Show-and-Tell is currently embedded in FORTRAN, we have written two sets of FORTRAN subroutines, SHAFT and SHAFTRANS.

1. SHAFT allows us to create, interactively, a data structure which represents a planar labelled cartoon, a cartoon being defined as a collection of labelled polygonal regions. As in Figure 1, the cartoon SEC1 consists of three regions A, B, and C. Each region consists of a list of pointers to a table of currently-used coordinates which define the vertices of that region.

Regions may be created either under program control (e.g. as output from a boundary-detection routine) or by the user using Show-and-Tell.

Additional hardware to enable the latter method (function switches and a hardwheel-driven cursor) is currently being constructed.

ii) SHAFTRANS allows affine transformations of a three-dimensional object defined by a set of lines. For example, a cube may be defined by four sets of lines as in Figure 2. The object thus defined may then be transformed and displayed on a plotter. Transformations currently supported are translation, rotation, scaling, and projection.

Figures 3 and 4 show a windowed view of a 50 x 50 grid, and two copies of that grid rotated 45°.

4.1.3 Terminal Communications

During the last quarter specifications were drawn up for a programmed I/O controller for LINC Tapes. The design of the controller is about 50% complete. The tape marking program and a program comparable to the DEC routine RWTAPE have been coded to make sure the control is usable.

The IBM 2701 interface has, at random times, missed the select and completion flags. The problem was found and will be corrected by adding two new IOT's: CLEAR DONE FLAGS and CLEAR SELECT FLAGS.

Due to budgetary constraints no work was performed on the multiple teletype network.

4.1.4 Scan/Display Devices

The flying spot microscope became operational during this quarter. Full control of X, Y, and Z axis motion is possible and under PDP-8e program control. X and Y steps move the slide by 34.4μm increments, while the Z-axis drive (focus) moves the slide by 0.12μm per step. The stage motors may be driven at up to 200 pps bidirectionally. Safety limit switches are not yet installed but will be. Users stand warned.

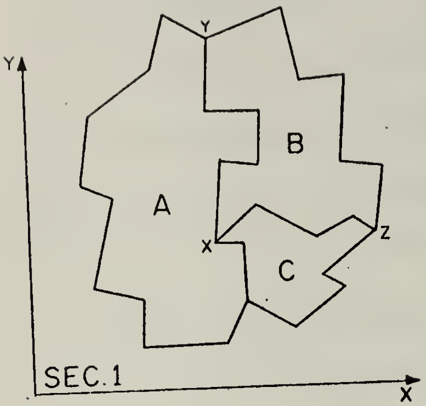


FIGURE 1.

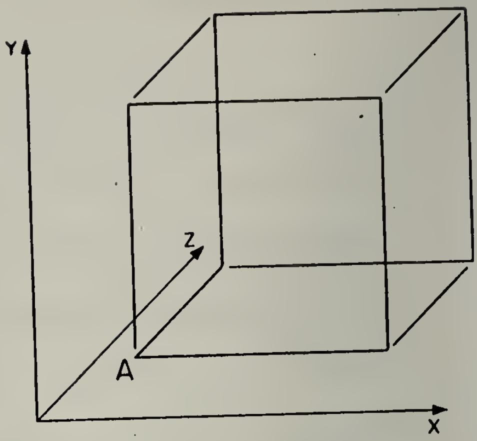


FIGURE 2.

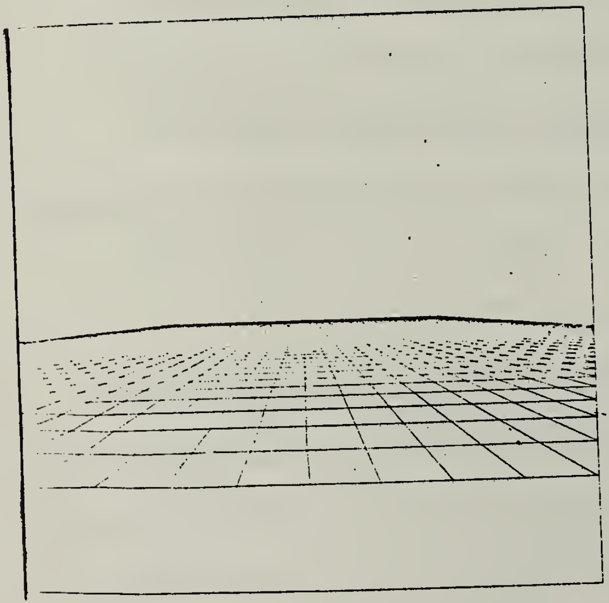


FIGURE 3.

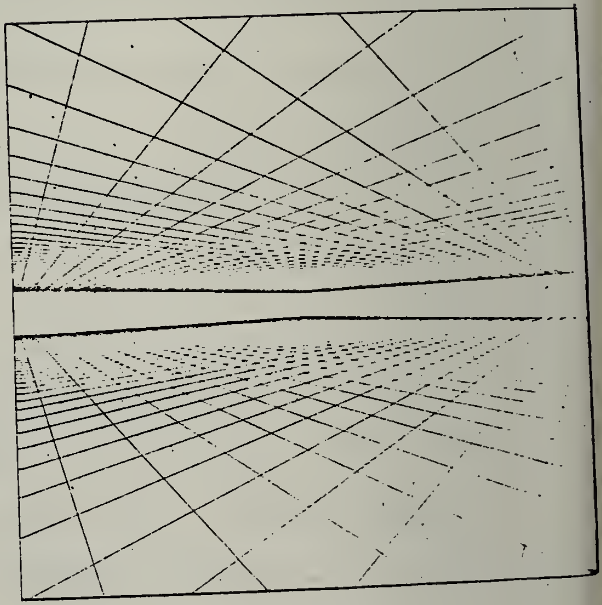


FIGURE 4.

A five lens turret will take any standard microscope objective (three powers are currently provided). A scheme was devised to add a reference photomultiplier before the spot image passed through the objective by using one of the optical charger parts. This was expedited, and an improved image was obtained since product detection was then possible. Resolution is very good but as yet unmeasured.

The CRT currently in use in this device is a Litton L-4238 with P-16 phospor. This tube is being operated at an excessively high beam current to overcome the light loss in the microscope optical system due to a small entrance pupil and to absorption of the CRT's blue UV light output by the large number of glass elements. It is anticipated that this situation can be improved by replacing the P-16 equipped CRT with a tube using one of the new broad-spectrum phospors; this will have the additional benefit of enabling experimentation with color data.

It was agreed that the abominable technique of reprinting 35mm film into 46mm strips could be eliminated by converting the spare 46mm transport into a sprocketed 35mm unit at minimal cost and effort. Plans for bringing up such a unit are in the wind. However, the new controller must be brought up to run the unit as this becomes the third operational scanner.

The second scanner, for use with video digitization techniques, is currently under construction. All outside contracted work for wiring of PCBs and back panel wiring have been completed. In-house additional back-plane wiring is in progress. Intra-drawer back-plane wiring is about 5% complete. Inter-drawer wiring is about 30% complete.

Four additional sections of the SMV manual have been completed. No additional work was done on the CIU or I/O sections.

4.2 PARALLEL PROCESSING

4.2.1 Extensions of Signal Detection Theory

The development of ROC technique for the analysis of texture has been described in earlier reports. This technique has been successfully used to extract textural regions, as described earlier. This can also be extended to extract textural borders in a combined scene, given the protosamples of the textures present in the scene.

The events extracted at the boundary where two textural regions meet in general are expected to be different from those that are extracted from the core of either region. This is so because these events are made up from the points from two different regions. We can say that the 'border texture' is different from either of the textures it is separating. Sometimes this may not be the case - the template may be too small to pick up these differences.

Let us consider the combined scene as T^1 and the union of all protosamples as T^0. We perform ROC analysis (Jayaramamurthy, S.N., "Application of Signal Detection Theory for the Analysis of Texture", not yet published) and obtain sets F^1, F^ϕ and F^0. Obviously F^0 is the null set because all the events in T^0 happen to occur in T^1 too. But the set F^1 contains the events that occur exclusively in T^1, which are nothing but the events extracted at the borders. We can go back and 'color' the conbined scene for F^1 alone and locate the border.

This technique has been applied to extract deformations in a textural scene, borders between natural textures, etc. A detailed account of this and other applications can be found in the reference cited above.

4.2.2 Interval Covering

Work on interval covers was resumed in mid-September. Experiments were done with the operational program used to determine interval covers which distinguish elements of a set F^1 from those of a set F^0 according to the technique presented in Michalski, DCS Report #450. Work was started on the problem of generating interval stars from explicit information about F^1 in the case where F^0 is too large to be worked with conveniently.

4.2.3 PAX Language Support

A second number of the PAX Users Group Newsletter was distributed this quarter to the thirty individuals returning a survey form included in the first issue. This survey revealed that as of July 1971, PAX II (or a variant) is operational at nine locations on four different computer types: CDC 3600, IBM 360, UNIVAC 1108 and PDP-10.

A name, "Plane Talk", for the PAX User's Group Newsletter has been arrived at by a somewhat democratic process.

Only a small number of minor corrections and modifications were made to the PAX II package this quarter.

4.3 STRUCTURAL INFERENCE

4.3.1 Scene Segmentation by Clustering

A careful review of published clustering techniques gave rise to the following observations:

1. A cluster is usually considered as a set of patterns in a multidimensional attribute space such that its density is large in comparison with neighboring sets of patterns. To measure the density of patterns in an attribute space, it is necessary to define at least a pseudo-metric in this space. No method to define a distance function between different attributes has been developed which is acceptable for scene analysis.

2. The clustering techniques available do not describe efficient algorithms to find clusters for the purpose of scene segmentation.

To overcome these difficulties, we are trying to develop a general model that allows us to define scene segmentation by clustering as an optimalization problem. To find an efficient clustering method, we are considering the following approaches:

1. Global approach: The overall density distribution of picture points is an attribute space used to find cluster centers in a scene.

2. Local approach: The cluster centers found in the global approach serve as centers for locally formed clusters using the nearest neighbor classification method.

4.3.2 SOL Language Support

This quarter the work centered around checking syntax of this graph-oriented language and preparing it for implementation. We have specified the root operations (mentioned in the previous Quarterly Progress Report) that are sufficient for expressing network transformations. They were prepared with the goal of embedding them in PL/1.

We have generated an algol program which differentiates SOL statements from PL/1 and checks for legitimacy of their syntax. At present we are attempting to implement the language with a two-pass compiler, where the first pass will replace SOL statements with their PL/1 equivalents and the second pass will actually be a PL/1 compilation.

4.4 APPLICATIONS

4.4.1 Cervical Smears

Development of this application has been completed to the limit of the current hardware/software tools. Using the PAX II package running under Show-and-Tell on the IBM 360/75 and PDP8/e, we have written programs which can perform attention-centering and some feature extraction on single, selected microscope fields where the degree of cell overlapping, clumping and degeneration is not too high.

These preliminary results were presented at two conferences, the Engineering Foundation's Research Conference on Automatic Cytology at Henniker, New Hampshire (July 26-28, 1971) and the Conference on Two-Dimensional Digital Signal Processing at Columbia, Missouri (October 6-8, 1971).

Work on refining and evaluating the programs will continue using the current facilities while the local image-processing capability is completed (See Section 4.1.1).

4.4.2 Brain Mapping

Clustering methods that are being developed for scene segmentation (cf. Section 4.3.1) will be applied to isolate nuclei of individual brain cells from background tissue as well as the delineation of boundaries in the gross anatomy of the brain.

Simultaneously we are developing optical filters for recognizing brain cells using the interval covering technique (cf. Quarterly Technical Progress Report, April-June 1971, Section 4.2.3.3).

A summary of our results will be issued later this year.

4.4.3 Cytospectrometer

During this quarter a number of schemes for particle (droplet) generation were investigated. Principally for the purpose of generating finer streams, and incidentally for avoiding clogging problems, a technique of forming a very fine-tipped cone of liquid (Taylor cone) on the order of a few tens of molecules radius by means of a high electrostatic field was tried. A number of runs with whole blood, fractions and bacteria were made onto glass slides with a crude device. Results were promising enough - single cell isolation is some cases - to justify the design of a better device. Construction of a six inch diameter modular vertical instrument is underway. Initial results with even the unstable jet were sufficiently interesting to suggest at least the temporary abandonment of the vibration type (Rayleigh jet) particle generator. Detection, stabilization and control schemes are proceeding apace.

4.4.4 Remote Manipulation

We have defined a class of objects as two-dimensional "angularly simple" polygons. Each such polygon has a convex set called its feasible set (Maruyama: unpublished, May 1971). To encode such a polygon into a unique radial pattern sequence, $S_r(x_0)$, we have developed computer procedures which first define a unique point x_0 in a given angularly simple polygon.

Our current interest in this class of polygons stems from problems of geometrical shape perception. From a psychological viewpoint (e.g. discrimination of an odd shape among many, detection of similarity or disimilarity between pairs of shapes, etc.) angularly simple polygons form a particularly attractive collection of shapes. The purpose of this study is to develop a computer system that gives solutions similar to those given by human subjects, and therefore provides a mechanism for naming objects in a scene in accord with given prototypes.

4.5 COMPUTER SYSTEMS

4.5.1 IBAL Assembler

The final version of syntax specification and language definition for IBAL (Illiac III Basic Assembler Language) has been completed and is reported in Report No. 469. IBAL is the basic language which will be used for all sophisticated system programs planned for the Illiac III system.

Bottom-up compilers are very attractive due to the fact that they provide a straight forward method for error recoveries. Considering Compiler Compiler systems we found TWS (Translator Writing System - Illiac IV) compatible with our requirements. This system generates a compiler by producing Floyd productions for syntax and linking the passer to different routines for semantic actions. Attempts to produce the first version of the compiler through TWS have been unsuccessful because after the generation of Floyd productions we found bugs in the later phase of the system, and unfortunately inadequate information was available to debug and retry the TWS.

We then tried to generate a top-down compiler by using the TWST system. We produced a syntax analyzer which is a long algol program. After making sure that the syntax was free of bugs, we are now using the 360 version of TWST called TACCS (A Table Driven Compiler Compiler System to build a compiler for IBAL.

In the first pass of the translator we deal mainly with syntax and data declarations and produce an intermediate code to be processed by the second pass. The structural tables and codes from the first pass are the input to the second pass, which generates the code for the actual memory allocation and pointer register allocation, and translates the intermediate code to actual machine language codes for Illiac III.

4.5.2 Operating Systems

Illiac III's operating system is described in the Illiac III Reference Manual, Volume IV: Supervisor Organization, which was printed during this past quarter on August 12, 1971.

The following concepts evolved during the past quarter:

1. In our multiprocessing environment, it is possible to have as many as four taxicrinic processors working on a given task. As was mentioned in a previous quarterly report, an INCK instruction prevented one interrupt from destroying the information peculiar to another interrupt. We have established that there will be a circular buffer of four pointers to available space to take care of each successively occurring interrupt. One extra space in this circular buffer will point to a first-in-first-out queue that stores these pointers. The queue is necessary when there are even more than four interrupts at one level, in which case the pointers will be overwritten and therefore lost if not stored in the queue.

2. In case further general interrupt information must be stored, provision has been made to allow storage in successive locations following the circular buffer.

3. All segments will be in a hierarchical structure. The structure is formed by using the accessing rights information stored in the Directory and in the Global Segment Table.

4.5.3 TM II

The following progress on the second generation Transfer Memory, TM II, was noted during the last quarter:

1. Design of PRE complete.
2. Design of discrete interface circuits between TM II and the IA complete.
3. Strategy for memory card and in particular chip selection complete.
4. Power supply (5v., 100A) ordered for evaluation.

The following tasks are expected to be completed during the next quarter:

1. Letting of bids for drawers and wiring for PRE.
2. Layout and fabrication of discrete interface cards.
3. Specifications for cabinet, cooling equipment, and necessary cabling.
4. Finalization of memory and card design.

5.4.4 TP, AU, and IOP

Work over the past quarter involved the publication of Volumes Two and Three (DCS Reports #475, 476) of the Taxicrinic Processor (TP) design. The former describes the basic machine control sequences and flow charts and the latter describes TP instructions. The design of the basic machine should be completed in the next quarter. Instruction sequence control logic design has been initiated. Completed instructions include those concerned with the operand stack, unary logical operations and list processing features of TP. A total of 183 TP control logic drawings have been completed. Of these, 77 have been mapped into completed cards and 51 are in the layout process. Thirty-one of the completed cards have final back panel wiring.

Documentation for the Volume Two of the Arithmetic Units (AU) describing the control sequences and their logical design was started and is continuing. It should be available in the coming quarter. A final index for the AU processing hardware and control logic drawings was prepared. 435 of 470 logical cards for processing hardware are in the machine after being checked out. Logic layout was completed for all of the 66 control logic cards and 15 more were wire listed and will soon go for wiring purposes.

Printed circuit board (discrete components) testing has been completed on 206 of the 250 boards required for the Input/Output Processor (IOP) processing hardware.

4.6 DOCUMENTATION

4.6.1 External Documents Issued

Report No. 469	Schwebel, J.C., Revised by Masumi, A.E., "IBAL MANUAL - The Illiac III Basic Assembler Language", July 1971.
UIUCDCS-R-71-470	Nordmann, B.J. Jr., "Speech Display Simulation System for A Comparative Study of Some Visual Speech Displays", August 1971.
UIUCDCS-R-71-472	McCormick, B.H., Nordmann, B.J. Jr., et. al., "ILLIAC III REFERENCE MANUAL - VOLUME IV: Supervisor Organization", August 12, 1971.
UIUCDCS-R-71-473	McCormick, B.H., Nordmann, B. J. Jr., et. al., "ILLIAC III REFERENCE MANUAL - VOLUME III: Input/Output", August 13, 1971.
UIUCDCS-R-71-475	Nordmann, B.J. Jr., "ILLIAC III COMPUTER SYSTEM MANUAL - Taxicrinic Processor, Volume 2", August 24, 1971.
UIUCDCS-R-71-476	Nordmann, B.J. Jr., "ILLIAC III COMPUTER SYSTEM MANUAL - Taxicrinic Processor, Volume 3", August 24, 1971.
UIUCDCS-R-71-479	Nordmann, B.J. Jr., "A Comparative Study of Some Visual Speech Displays", September 10, 1971 (Ph.D. Thesis).
File Note No. 863	Borovec, R. T., "IMAGE 8: The Real Time Clock", June 16, 1971.
File Note No. 864	McCormick, B.H. (ed.), "Illiac III Bibliography Condensed Listing - Supplement" , July 13, 1971.
File Note No. 868	Lewis, G.T., "IMAGE 8: Illiac Paper Tape Reader Control", August 10, 1971.

4.6.2 Logic Drawings Issued

The following new logic drawings were issued uring the past quarter:

TP Control Logic	30 drawings
AU Control Logic	17 drawings
Power Turn-on Logic	20 drawings

4.6.3 Engineering Drafting Report

During the past quarter a total of 459 drawings, including new logic drawings, drawing changes, layouts, flow-charts, theses, report drawings and drawings related to the Opto/Mechanical design of the Illiac III project have been processed by the 2118 drafting section.

4.7 ADMINISTRATION

4.7.1 Personnel Report

Senior Staff

Professor Bruce H. McCormick - Principal Investigator
Assistant Professor R. S. Michalski

Professional Staff

Robert C. Amendola
Richard T. Borovec
John S. Read

Research Engineering Assistant

S. Paul Krabbe

Electronic Engineering Assistant

Joseph V. Wenta

Digital Computer Technician II

George T. Lewis

Drafting

Stanislavs Zundo

Research Assistants

Jerry Chen
Walter Donovan
Steve Fierce
Lakshmi Goyal
S. N. Jayaramamurthy
Ahmad E. Masumi
Kiyɔshi Maruyama
Frank Murakawa
Dan Pitt
Peter Raulefs
Kuo Wen

Secretarial

Mrs. Judy Arter

4.7.2 Computer Usage Log Summaries

1. The PDP/8e was used approximately 250 hours.

2. The Scanner-Monitor-Video System was used as follows:

Production and Demonstration	129 hours	35 minutes
Preventive Maintenance	12 hours	10 minutes
Corrective Maintenance	27 hours	0 minutes

3. The IBM/75 utilization totaled $3,878.85.

Form AEC–427
(6/68)
AECM 3201

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY–TYPE CONTRACTOR'S RECOMMENDATION FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.
COO-2118-0025

2. TITLE
Illiac III Quarterly Progress Report

3. TYPE OF DOCUMENT (Check one):

☒ a. Scientific and technical report
□ b. Conference paper not to be published in a journal:
Title of conference ______
Date of conference ______
Exact location of conference ______
Sponsoring organization ______
□ c. Other (Specify) ______

4. RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

☒ a. AEC's normal announcement and distribution procedures may be followed.
□ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
□ c. Make no announcement or distrubution.

5. REASON FOR RECOMMENDED RESTRICTIONS:

6. SUBMITTED BY: NAME AND POSITION (Please print or type)
Bruce H. McCormick
Professor of Computer Science and Physics

Organization
Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature Bruce H. McCormick

Date 12/1/71

FOR AEC USE ONLY

7. AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION RECOMMENDATION:

8. PATENT CLEARANCE:

□ a. AEC patent clearance has been granted by responsible AEC patent group.
□ b. Report has been sent to responsible AEC patent group for clearance.
□ c. Patent clearance not required.

5. ILLIAC IV

(This work was supported in part by the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.)

REPORT SUMMARY

The contract between Automation Technology Incorporated and the University of Illinois expired during the quarter. Under its new contract the Center for Advanced Computation has prime responsiblity for applications on the ILLIAC IV. Progress will be reported under Contract No. DAHC04-72-C-0001 monitored by the United States Army Research Office-Durham.

The ILLIAC IV software neared completion this quarter. Users are in a position, via the ILLIAC IV simulator, to get first hand experience with all the ILLIAC IV languages and library routines in all areas except the operating system. The operating system has been integrated with the ILLIAC IV simulator, and all modules have been tested in concert. A more sophisticated version is being developed that will allow simulation using the actual I/O subsystem when it becomes available.

The operation of the B6500 installation at the University of Illinois has improved considerably during the quarter. The system is now available approximately ninety percent of the time.

Project expenditures through September 1971:

Burroughs Corporation	$27,159,707.00
University of Illinois	8,257,550.00*

*Estimate.

HARDWARE

The contract between Automation Technology Incorporated and the University of Illinois expired during the quarter. No future hardware reports will be made.

SOFTWARE

5.1 Operating System

The three major Operating System activities progressed well during the quarter. The first, integration of the ILLIAC IV Operating System with the ILLIAC IV Simulator, was successfully completed in July and has been undergoing refinement since then. This integration is known locally as the "Link Trainer," mainly because it allows users to submit ILLIAC IV jobs to the B6500 just as if they were submitting them to the B6500-ILLIAC IV constellation. Even though there are several restrictions, a number of test cases were successfully run with ASK and GLYPNIR programs. During the course of integration a number of minor incompatibilities within the operating system were found and corrected. Work is now proceeding on the next stage of integration — the "Steerman Trainer," which, although still using the simulator, will exercise the ILLIAC IV I/O subsystem. This work is proceeding well but will soon be blocked by the unavailability of the I/O subsystem at Paoli.

Secondly, the interface between the University Operating System and the Burroughs on-line diagnostics was defined and is being implemented by both parties.

The third system, the interface with the debugging package being developed by Bolt, Beranek and Newman, has been designed. Both the University and Bolt, Beranek and Newman are working on modifications to OS4 (the ILLIAC Resident Module of the Operating System) to accomodate this package.

Preparations are under way to transfer from the B6500 to ILLIAC IV some modules of the Operating System whose algorithms are inherently parallel. This work is being done so that these changes can be made as soon as the present Operating System has been checked out on the hardware.

5.2 COCKROACH

The virtual completion of a production version of the COCKROACH-to-GLYPNIR translator and its associated documentation was accomplished. Much time was spent improving error detection and recovery facilities in the translator. Various storage allocation studies for arrays of row dimension greater than 64 were investigated with a view to selecting one which provides efficient calculation and ease of column and plane indexing for implementation into the translator. Implementation (with some restriction) occurred for array storage and manipulation facilities for arrays of any dimension. Straight storage and physical skewing were selected after considering the results of the storage allocation study. Binary vectors longer than 64 were executed. These provided mode vectors for the array scheme.

The loop variable was used in place of the mode asterisk. At the suggestion of users the construction

DP #1 = BINARYVARIABLE

was admitted into COCKROACH. This removes the need to repunch parallel sections of the code. The provision of a COCKROACH-to-FORTRAN translator on the B6500 allows users to develope parallel algorithms in COCKROACH without the high overhead of simulating on the ILLIAC IV Simulator. This work is now nearing completion. The COCKROACH translator was translated from Burroughs Compatible ALGOL into Burroughs Extended ALGOL.

5.3 GLYPNIR

Maintenance and improvement of the GLYPNIR compiler continued over the quarter. There were three main activities: 1) The compiler was translated from Burroughs Compatible ALGOL to Burroughs Extended ALGOL. This raised its compile speed from about 200 to about 350 cards per minute. 2) Work continued on the new and faster scanner for GLYPNIR. This will also provide macro facilities for the language. 3) ILLIAC (as opposed to simulator) I/O facilities are being added to the language, and work is progressing well in this area.

Because the GLYPNIR compiler is being used by many and is working well, its development was slowed down. Some of the GLYPNIR team were temporarily re-assigned to more urgent work such as managing the B6500 installation and helping with Operating System development.

5.4 ILLIAC IV Assembler (ASK)

The Assembler underwent considerable refinement over the quarter. Its present peak compiling speed for GLYPNIR generated code is 2300 cards per minute. It was made into a standard B6500 compiler, and facilities were added to allow a "compile and simulate." No significant changes to the language itself were made during the speedup.

5.5 Documentation

Reference and maintenance manuals for all major software modules developed by the University were completed during the period. In addition, introductory manuals were provided to help people become acquainted with the B6500 and the ILLIAC control language. They are available for distribution.

5.6 Burroughs B6500 Facility at Illinois

The performance of the B6500 installation at the University of Illinois improved considerably during the quarter. In mid-July the air-conditioning system was settled into its summer regime and since then has given little trouble.

In the beginning of August level 1.10 software was used. This checked the rising rate of "Halt Loads" (when the processor stops and has to be re-initiated by hand). Since then the Halt Load value has remained constant at about 100 a week. The middle of August saw a marked improvement in hardware faults. The average amount of downtime attributable to hardware malfunction was less than 8 hours a week.

Two extra memory modules of 16K words each have been obtained. One was connected in mid-September with resulting improvement in machine performance. At present about 10,000 jobs are processed weekly by the B6500. Detailed accounting procedures will be implemented on the machine in the near future.

APPLICATIONS

The Center for Advanced Computation of the University of Illinois at Urbana-Champaign now has responsibility for applications on the ILLIAC IV. Progress will be reported under Contract No. DAHC04 72-C-0001 monitored by the United States Army Research Office-Durham.

5.7 Financial Report

Expenditures and obligations for July, August and September 1971

	July-Actual	August-Estimate	September-Estimate
Burroughs	$ 26, 265.00	$155,397.00	Not available
University	$289,996.00	$121,000.00	$55,000.00

Estimated expenditures and obligations through September 1971

Burroughs Corporation	$27,159,707.00
University of Illinois	$ 8,257,550.00

THESES

Kelly, M. W. "Reliability of Computer Systems." Master's Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

Miura, Ken'ichi. "Parallel Radiation Transport Code: A Monte Carlo Method Applied to ILLIAC IV and Its Statistical Consideration." Master's Thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

DOCUMENTS

Abel, N. "A FORTRAN-based Language for Array Processing." ILLIAC IV Document No. 246, ILLIAC IV Project, University of Illinois at Urbana-Champaign (March 30, 1971).

Arnold, C. "Roth's Algorithm for Diagnostic Test Generation: Theory and Implementation." ILLIAC IV Document No. 240, ILLIAC IV Project, University of Illinois at Urbana-Champaign (January 22, 1971).

Knapp, M. and D. McIntyre. "Bulk Storage Applications in the ILLIAC IV System." ILLIAC IV Document No. 250, ILLIAC IV Project, University of Illinois at Urbana-Champaign (August 3, 1971).

Nakamoto, H. and A. Hashimoto. "Optimal Frequency for Hardware Confidence Testing." ILLIAC IV Document No. 241, ILLIAC IV Project, University of Illinois at Urbana-Champaign (January 22, 1971).

S. T. "ICL Your Fancy." ILLIAC IV Document No. 252, ILLIAC IV Project, University of Illinois at Urbana-Champaign (September 1, 1971).

6. SWITCHING THEORY AND LOGICAL DESIGN

(Supported in part by the National Science Foundation under Grant Number U.S. NSF-GJ-503.)

The branch-and-bound method finds an optimal network early during computation but proving the optimality of that network takes the majority of computation time. So we tried to find procedures to find an optimal network as early as possible. If such procedures are available, a reasonably good network could be obtained even if a computer is halted. New procedures which appear to be more powerful than any currently known were devised and we started to program them. How powerful they could be is to be seen soon.

Optimal one-bit adders were further designed for additional combinations of different types of gates.

S. MUROGA

In order to obtain a near-optimum network of NOR gates, one approach is to transform a network into one of less cost without altering the output functions. We developed five such transformation procedures.

I: (1) A sufficient condition to detect redundant connections and gates in a network.

(2) A necessary and sufficient condition to detect redundant connections and gates in a network.

II: A procedure for transforming a network by adding connections and then removing different connections.

III: A procedure to remove more than one connection or gate simultaneously.

IV: A procedure to remove one gate using I (1) and III.

V: A procedure to remove one gate (a generalization of the procedures of Ellis).

Procedure I (1) is a generalization of a previous procedure developed by our group.

Although these procedures appear more effective than previously known transformation procedures, further modifications and generalizations may be necessary.

(Y. Kambayashi)

The study of network transformations by which a non-optimal network of gates is reduced to a (hopefully) near-optimal network (while preserving the original output functions) was continued.

Procedures for five such transformations were outlined by Kambayashi. Of these, Procedures II, IV and I (1) were to be programmed by me. Programming is nearly completed, with Procedure II apparently debugged and operating.

Not enough results have been obtained so far to permit a reasonable discussion of the effectiveness of these procedures.

(J. N. Culliney)

Two procedures, procedures I and V, to transform a given NOR network to a network with lower cost were coded into the following three FORTRAN subroutines:

RDTCNT: Remove all single-redundant connections by necessary and sufficient conditions.
An entry point REMCNT (GY, GZ) was included in this subroutine. By calling this entry point the connection from GY to GZ will be examined to see whether it is removable or not.

PROCV: Check whether a gate GI can be removed or not by adding connections to the gates originally fed by GI.

RQRNW: Calculate requirements for each gate except GI, and for each connection from GI. This subroutine will be called by PROCV.

A main program to read in NOR networks and to transform them to networks with lower cost by calling the above two procedures was coded.

(H. C. Lai)

The synthesis problem of networks for negative functions by MOS cells with a minimum number of FET'S was continued. Some preliminary results on the synthesis of optimal MOS networks for multi-output functions were obtained.

The design of optimal one-bit full adder with different types of gates was considered again. In addition to the four types of networks in Report 425, optimal networks of nine different combinations of restrictions were obtained and cataloged.

(T. K. Liu)

Report No. 455

Nakagawa, Tomoyasu and Saburo Muroga, "Comparison of the implicit enumeration method and the branch-and-bound method for logical design," Departmen of Computer Science, University of Illinois, June 1971, 94 pages. (The implicit enumeration method and the branch-and-bound method are compared in terms of ease of programming and computation time when they are applied to logical design. Generally the latter is faster in computation time but more time-consuming in programming than the former.)

Report No. 462

Nakagawa, Tomoyasu, "A branch-and-bound algorithm for optimal AND-OR networks (The algorithm description)," Department of Computer Science, University of Illinois, June 1971, 39 pages. (A branch-and-bound algorithm to derive optimal networks of AND and OR gates for a given switching functi is described.)

Report No. 471

Nakagawa, Tomoyasu, Hung Chi Lai, and Saburo Muroga,"Pruning and branching methods for designing optimal networks by the branch-and-bound method," Department of Computer Science, University of Illinois, August 1971, 59 pages. (When optimal networks are designed by the branch-and-bound method, an optimal network is found early during computation but proving its optimality consumes major part of the computation. In this paper some procedur to further speed up the finding of an optimal network are discussed.)

Report No. 480

Culliney, Jay Niel, "On the synthesis by integer programming of optimal NOR gate networks for four variable switching functions," Department of Computer Science, University of Illinois, September 1971, 57 pages. (Optimal networks of NOR gates are designed by the implicit enumeration method for some functions of four variables.)

Operations Research, Vol. 19, No. 4

Baugh, C.R., T. Ibaraki, and S. Muroga, "Results in using Gomory's all-integer algorithm to design optimum logic networks," July-August 1971, pp. 1090-1096. (Optimal networks of NOR gates are designed by Gomory's all-integer integer algorithm. Computational efficiency for this logical design problem is discussed.)

7. SOUPAC

(Statistically Oriented Users Programming and Consulting)

July - September 1971

SCANSOUP to codecheck a SOUPAC program on Express is available. Several new programs also became available to users:

MULTIVARIATE PROBIT and the REGRESSION Package,

ORTHOGONAL PROCRUSTES, VIEWPOINTS and VARISIM for factor analysis,

CROSS-SPECTRAL ANALYSIS for autocorrelations, and

MANN-WHITNEY for nonparametric statistics.

Documentation for these programs and revised write-ups for roughly 50% of the older SOUPAC programs will appear in the next SOUPAC manual.

A procedure called TESTSLOT which allows any functioning FORTRAN program to be temporarily incorporated into SOUPAC has long been available. Recently this technique was used by a Commerce class to run a specialized routine in connection with the statistics of SOUPAC.

A SOUPAC consultant has been keeping regular office hours at Commerce. The purpose is twofold: to help the many Commerce users with their SOUPAC problems near their terminal and to interact with Commerce personnel on questions of mutual concern. A Seminar with Commerce faculty on SOUPAC was scheduled for October.

Of the new programs added to SOUPAC this quarter, the Regression-Correlations Package is likely to have the most far-reaching consequences. This program was conceived and written so that users desiring a number of statistics derived from the same correlation matrix or a subset of that matrix could obtain their results more efficiently. Even in SOUPAC, a series of calls to the same program, each one involving a calculation of of the correlation matrix, might be necessary to do several regressions from the same data. The recalculation has now been eliminated. Matrix manipulations to select variables are also not necessary. A single statement executed under the control of Regression selects and reorders variables as needed. The programs available under Regression are Simple, Multiple and Stepwise Regression, and Partial and Canonical Correlation. The previous versions of these programs are still available for users who do not require the power of Regression.

SOUPAC usage during the period was more than 3600 jobs per month or 117 jobs per day, although useage in September was less than half that of July due to vacations.

October 3 0, 1971
JM;jk

8. MACHINE AND SOFTWARE ORGANIZATION STUDIES

(Supported in part by the National Science Foundation under Grant Number US NSF GJ 27446)

The following is a collection of related work aimed at improved designs for computer and software systems. We are interested in parallel processors, small primary memories, effective use of rotating memories and some questions concerning user languages for problems including typical Fortran type calculations and certain file processing problems.

8.1. Fortran Parallelism Detection - (S. Chen)

The programs to extract parallelism have been largely rewritten. The old scanner generated the tables of information for the DO-Loop subprogram in such a way that the analysis of DO loops had to start from the innermost independent blocks and progress outward treating each new block independently. This strategy might inhibit some potential parallelism which could otherwise be exploited by combining the inner blocks with the outer one. For example, the following structure can be transformed to the equivalent structure without effecting its logic, but with more parallel operations found by the new extracting programs.

```
DO I=1, 10                        DO I=1, 10
A(I)=B(I)                         A(I)=B(I)
DO J=1, 10                        DO I=1, 10
C(I,J)=A(I,J)+D(J)                DO J=1, 10
                                  C(I,J)=A(I,J)+D(J)
```

After scanning a real program, the new extracting programs use a DOHANDLE routine to keep the list structure of the original DO loop, and rearrange the combination of blocks before its actual analysis of parallelism.

This also leads to reducing a lot of effort (machine time and storage) spent in the old DO-Loop subprogram in order to "synthesize" the block structure from the fixed table information supplied by the old scanner.

In response to this new strategy, the DO-Loop subprogram has been greatly simplified. Now it serves only as an essential part of the DOHANDLE routine, performing the function of finding possible separations within any nested DO loop and evaluating the parallelism in a single statement block after the proper combination has been found.

The old DO-Loop subprogram has been discarded and the new version is integrated into the DOHANDLE routine of the new extracting programs, which have been tested successfully for many real FORTRAN programs. The next step will be the incorporation of the assignment subprogram ASSIGN with its new backsubstitution and recurrence features.

8.2. Tree Machine Simulator - (P. Budnik)

A program to simulate the arithmetic unit and memory of a highly parallel and very flexible computer has been written and debugged. It has been used to estimate execution time for several sections of FORTRAN programs. We plan to expand this program further, and to test different hardware configurations as well as run many more experiments with different FORTRAN programs.

8.3. Weighted Node, Directed, Acyclic Graph Schedule by m-Machines (P. Kraska)

During the previous quarter, theorems were developed to parse arithmetic expressions such that tree-height is minimized when the expression consists of a sum of terms or a product of factors, and the operator weights are not uniform. These expressions may be represented by:

$$f = \sum_{i=1}^{n} t_i \quad \text{and} \quad t = \prod_{i=1}^{m} f_i ,$$

where Σ implies addition or subtraction and Π implies multiplication or division. Furthermore, if we permit distribution then an algorithm which reduces the tree-height of t was shown to be valid.

A PL/1 program was written during this quarter which accepts a FORTRAN statement, e=f/t, as input, and with given operator weights, parses e such that the tree-height of e, h(e), is reduced for all cases except when e becomes $e'=(\overset{n}{\Pi} f_i)/(\overset{m}{\Pi} f_i')$, where $m > 1$. This case will be completed during the next quarter. We also include memory access (fetch and store) as an operator and we modify expressions of the form $e=-t_1-t_2-\ldots-t_n$ to $e'=e+0$ to eliminate the unary minus problem.

8.4. Simulation Processor - (E. Davis)

A system is being programmed to evaluate performance of the simulation machine. This includes a GPSS simulator of the machine, a PL/1 program to prepare test simulation programs for input, a 360 assembly language routine to extract run time parameters from test programs, and an assembly language routine to insert the parameters into the model when running tests on the simulator. The extraction and insertion routines provide information related to an actual run so that the same execution routines will be simulated. In the next quarter the system should be completed, then experimentation and evaluation can begin.

For those parts of simulation programs where the ratio of decision statements to assignment statements is high, an algorithm has been written which concentrates the decisions into a binary tree free of assignments.

Hardware has been designed for the specific purpose of evaluating the tree. A program has been written which converts FORTRAN arithmetic IF statements into logical IF's which have the two way branches necessary for the tree. Design effort will continue on the interface between the decision evaluator hardware and the processors.

8.5. Memory Hierarchies - (D. Gold)

The collection of algorithms which operate on FORTRAN-like programs to produce latency-free I/O mappings for small primary memory systems have been largely formalized. They are now being committed to paper in a comprehensible form. Furthermore, a program which obtains the I/O patterns for random perturbations of data partitions has been debugged, thus verifying the algorithm.

9. COMPUTER SYSTEMS ANALYSIS

(Supported in part by the National Science Foundation under Grant No. US NSF GJ 28289.)

The goal of this research is the development of analytical tools for system modeling and analysis of real time computer networks. A queueing theory model for a geographically distributed computer network is being developed. Priority assignment and job dispatching rules for efficient job processing in the network are being investigated.

9.1. Computer Network Modeling (W. McKinney)

We have formally defined the queueing theory model our multiserver system. The model includes J job types and K processors. Each job type has its own arrival rate λ_j and its own queue. Each of the J queues has its own length m_j and ρ_j priority classes. The processing rates for the K processors are given in a matrix μ_{jk} which gives the processing rate of type j jobs on processor K. The use of a processing rate matrix enables us to have identical or heterogeneous processors, and allows each processor to have preferred job types.

To date we have considered nonpriority, nonpreemptive priority, and preemptive priority disciplines. We are currently developing a GPSS model to facilitate studying the effects of priority assignment, job dispatching, and load regulation within the network.

9.2. Center Throughput Analysis (S. Mamrack and W. Barr)

We are currently investigating an algorithm which minimizes the maximum (nonlinear) cost function in throughputting n jobs on 1 computer when the jobs arrive simultaneously and have no precedence constraints. A dynamic programming approach has been successful in minimizing the cost function, but involves a prohibitive number of calculations for large n. Approximation techniques developed so far do not guarantee approaching optimality. We plan to combine the linear programming and branch-and-bound techniques to reduce the number of calculations by an order of magnitude, and still guarantee optimality. If this work is successful for the n/1 problem we will extend the results to the more general case.

We are also developing a new measure of goodness for evaluating center throughput. We define a measure of goodness, γ, as follows:

$$\gamma = \left(\frac{L_q}{M}\right)^{\alpha} \left(1 - \frac{M-L_q}{\sum_{i=0}^{R-1} \beta(i)}\right.$$

where

L_q is the number of jobs in the queue

M is the maximum length queue

R is the number of priority classes

and

α and β(i) are control parameters.

We are considering a family of β(i)'s, ranging from the nonpriority case to the preemptive dynamic priority case. After having established the γ function as a desirable measure of goodness, we will relate the α's and β's to the center's control parameters which are under the designer's (user's) control. Then, we plan to extend the technique to determine throughput values for a geographically distributed network of computers. The existence of the γ measure will provide us with the capability to compare centers within a net and hence to quantitatively determine criteria for job dispatching between centers.

(E. K. Bowdon)

10. THEORY OF DIGITAL COMPUTER ARITHMETIC

(Supported in part by the National Science Foundation under Grant Number US NSF GJ 813).

10.1. Continued Fraction Arithmetic

A study of the general use of continued fractions in digital computer arithmetic was begun during the quarter. The main goal is to find the class of functions that can be efficiently evaluated by the use of continued fraction methods.

Let f_1 be a given function, and let

$$f_1 = a_0 + \frac{b_1}{a_1} + \frac{b_2}{a_2} + \ldots + \frac{b_n}{a_n} .$$

Then the recursion

$$\begin{aligned} &P_i = a_i P_{i-1} + b_i P_{i-2} \\ (1)\quad &Q_i = a_i Q_{i-1} + b_i Q_{i-2} \qquad i = 2, 3, \ldots n, \\ &P_0 = 0,\ P_1 = b_1,\ Q_0 = 1,\ Q_1 = a_1 \end{aligned}$$

defines uniquely f_1 as $f_1 = P_n / Q_n$.

The computational algorithm of (1) involves four multiplications for each iteration. To reduce this number, simple continued fractions were considered, i.e. $b_i = 1$, $i = 1, 2, \ldots n$. Furthermore we hope that a_i can be simplified in the binary sense, i.e. $a_i = 1$, 1/2 or 1/4 for all i.

Our research began with a study of simple continued fraction presentations of functions as they appear in the literature. The a_i are assumed to be integers, and in many cases they include a variable x.

Example:

$$\tan x = \frac{1}{\frac{1}{x}} - \frac{1}{\frac{3}{x}} - \frac{1}{\frac{5}{x}} - \cdots\cdot$$

The result is that algorithm (1) requires at least two multiplication operations for each iteration, hence this approach was abandoned.

Our new approach is to attempt to develop new algorithms in order to avoid the high number of multiplications.

Amnon Bracha

10.2 Continued Fraction Algorithm for the Square Root

An algorithm to solve the set of quadratics $x^2 + b_k x - c_k = (x-u)(x-v) = 0$ with $b_k = 0$ (i.e., the square rooting problem) is now available. A proof of convergence of the algorithm has also been obtained.

Here u is to be expressed in the form of an infinite continued fraction,

$$u = \frac{1}{q_1} + \frac{1}{q_2} + \quad \cdots \quad + \frac{1}{q_i} \quad + \quad \cdots$$

where $q_i \in \{1/4, 1/2, 1\}$.

Recursion relations were developed earlier and have been reported.* The problem remaining was to determine the rules for selection of q_i. These rules have now been found.

Assuming that we are dealing with floating point numbers, the mantissa of c_k will be in the range $[1/2, 1]$. But since the exponent can be either odd or even and since for square rooting, we need to have an even exponent, conditional unnormalizing by 1 bit is necessary. Thus the range of the mantissa of c_k doubles. Either $[1/4, 1]$ or $[1/2, 2]$ can be chosen. In this algorithm the former choice has been made. As a result $1/2 \leq u \leq 1$.

Kishor S. Trivedi

*Quarterly technical progress report, Department of Computer Science, University of Illinois, April-June, 1971, pp. 142-143.

10.3. Evaluation of Some Elementary Functions in Radix 16

During this quarter some properties of previously obtained algorithms have been studied. One interesting feature of these algorithms, which are essentially based on a continued product representation of numbers, is the possibility of predicting larger and larger numbers of successive coefficients (s_k) at each step. This represents one way of achieving faster convergence of iterative processes, provided that a corresponding increase is made in the number of levels in a stacked adder structure.

The chosen algorithms have also been checked and the formalization of the results in the form of an M.S. thesis is in progress.

M. D. Ercegovac

Publications

Report No. UIUCDCS-R-477

Diamond, Martin, "A factorization procedure that is equivalent to successive over relaxation," Department of Computer Science, University of Illinois, September, 1971.

Report No. UIUCDCS-R-478

Diamond, Martin, "The divergence of Stone's factorizations when no parameters are used," Department of Computer Science, University of Illinois, September, 1971.

11. COMPUTER LANGUAGES FOR MATHEMATICS AND NUMERICAL ANALYSIS

We present abstracts of Digital Computer Laboratory reports and papers which have appeared during the last six months. This work has been supported by the National Science Foundation under NSF Grant GJ-328.

11.1. COMPUTER LANGUAGES:

Phillips, J. Richard, H. C. Adams, "Dynamic Partitioning for Array Languages. (To appear in CACM.)

ABSTRACT: The classical process of partitioning an array into subarrays is extended to a more useful array language operation. Various modes of partitioning are defined for different types of arrays, so that subarrays may vary over the original array in a nearly arbitrary manner. These definitions are motivated with several realistic examples to illustrate the value of partitioning for array languages.

Of general interest is the data structure for partitioning. This consists of dynamic tree structures which are used to derive and maintain the array control information. These are described in sufficient detail to be of value in the design of other array languages. The description presented in this paper is implemented in a new array language, OL/2, currently under development at the University of Illinois.

11.2. NUMERICAL ANALYSIS:

Saylor, Paul E., "Second Order Strongly Implicit Symmetric Factorization Procedures for the Solution of Elliptic Difference Equations". (To appear.)

ABSTRACT: H. L. Stone has proposed an iterative method, called a factorization procedure, to solve elliptic difference equations. Because of the complexity of the algorithm defining the iteration matrix, Stone described the procedure as strongly implicit. Analysis of its convergence properties is difficult. One reason is that the iteration matrix is non-symmetric. This motivates the construction of symmetric strongly implicit factorization procedures. In this paper the class of symmetric strongly implicit factorization procedures is determined. From this, it is possible to determine the class of such procedures that are second order. This is a property of the Stone procedure that seems fundamental to its success. Since the iteration matrix of any of the resulting procedures is defined by an algorithm with unpleasant properties, no satisfactory second order symmetric strongly implicit factorization procedure exists.

Bracha, Amnon, "A Symmetric Factorization Procedure for the Solution of Elliptic Boundary Value Problems," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 440, April, 1971.

ABSTRACT: The convergence properties of a symmetric factorization procedure for solving elliptic difference equations due to H. L. Stone is studied. Convergence depends on bounds on a parameter, and computable values of these bounds are obtained. The procedure and the results are generalized to an arbitrary number of dimensions.

11.3. ALGORITHMS:

Maruyama, Kiyoshi, "Parallel Methods and Bounds of Evaluating Polynomials," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 437, March, 1971.

ABSTRACT: A lower bound of the maximum evaluable degree of polynomial for each step s is given, and is $N^*(s) > 2^{s(1-\delta)}$, $\delta \approx (2/s)^{1/2}$. An upper bound is derived for the number of steps required to evaluate polynomials of degree n, and is given by $T(P_n) < (1+\epsilon)\log_2 n$, which approaches theoretical lower bound $\lceil \log_2(2n+1) \rceil$ as $n \to \infty$, where $\epsilon \approx (2/\log_2 n)^{1/2}$.

An upper bound of the number of operations required to achieve the theoretical lower bound of steps required to evaluate polynomials of degree n is given as Cn, C> 2. Furthermore a systematic construction of computation trees, multi-folding and modified multi-folding methods, which achieves the bound is given. Our dual problem approach leads to better results than Brent's [1] primal problem approach, and shows the existance of a simple scheduling algorithm to evaluate polynomials within the bound.

Maruyama, Kiyoshi, "A Procedure for Detecting Intersections and Its Application," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 449, May, 1971.

ABSTRACT: The procedure described here employs face-to-face intersection analysis to determine whether two or more polyhedral objects intersect. As means to minimize the number of pairs of faces which should be examined for face-to-face intersection analysis, a solution box approach, mutual divisibility and visibility of two faces are considered. Intersection detection between two faces is done by the determination of their parity mode.

As an application of the procedure developed, a solution for the path finding problem in geometrically constraint space is given.

REFERENCES

Bracha, Amnon, "A Symmetric Factorization Procedure for the Solution of Elliptic Boundary Value Problems," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 440, April, 1971.

Maruyama, Kiyoshi, "Parallel Methods and Bounds of Evaluating Polynomials," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 437, March, 1971.

Maruyama, Kiyoshi, "A Procedure for Detecting Intersections and Its Application," Department of Computer Science, University of Illinois at Urbana-Champaign, Report No. 449, May, 1971.

Phillips, J. Richard, H. C. Adams, "Dynamic Partitioning for Array Languages. (To appear in CACM.)

Saylor, Paul E., "Second Order Strongly Implicit Symmetric Factorization Procedures for the Solution of Elliptic Difference Equations". (To appear.)

12. NUMERICAL METHODS, COMPUTER ARITHMETIC AND ARTIFICIAL LANGUAGES

(Supported in part by the National Science Foundation under Grant No. US NSF GJ 812.)

12.1 An Adaptive Algorithm for the Solution of Finite Difference Equations

By modifying the algorithm we developed last quarter, we have obtained a second adaptive algorithm for the solution of large systems of linear equations. Both algorithms solve the system $Ax = q$ by using the factorization iteration

$$(A+B)x_{n+1} = (A+B)x_n - \tau_n (Ax_n - q),$$

where B is the auxiliary matrix defined by the factorization. They are adaptive in the sense that they generate the parameters $\{\tau_n\}$, from quantities computed on previous iterations, so that the rate of convergence approaches the rate obtained when the optimal set of parameters is used.

The sequence of parameters is made up of a number of subsequences, $\tau^i_0, \tau^i_1, \ldots \tau^i_{N-1}$, $i = 1, 2, 3, \ldots$, each defined as a Tchebychev sequence on an interval (a_i, b_i). In the first algorithm, the interval (a_i, b_i) is used to define $x^i_0, \ldots x^i_{N-1}$. Then after determining the improved interval and defining $x^{i+1}_0 = x^i_{N-1}$, the iterates $x^{i+1}_1, \ldots, x^{i+1}_{N-1}$ are calculated.

In the modified algorithm, x^{i+1}_0 is defined as a linear combination of $\{x^i_n\}^{N-1}_{N=0}$ so that x^{i+1}_0 equals the vector that would have been calculated if the improved interval had been used in the last N iterations. This modification increases the rate of convergence significantly since the iterates $x^i_1, \ldots x^i_{N-1}$, which are originally calculated without any knowledge of the optimal interval, are combined to yield x^2_0 determined by the interval (a_2, b_2) which is derived using the knowledge gained in the first N-1 iterations.

(M. Diamond)

12.2 ETS Operating System

Progress on ETS (An Educational Timesharing System for the PDP-11) has been slowed somewhat by assorted hardware/software/human problems; however, some major check points have been reached.

The timesharing monitor is essentially debugged. It has supported up to 8 consoles interactively for extended periods in test situations. The monitor filing system is also essentially debugged. Some new non-resident service routines are still to be added which will be service modules for specific system software running under the monitor.

A new command string interpreter has been written and is in the debugging stage; it is to replace the present "test only" command string interpreter which is both sensitive to user errors and requires an understanding of the system beyond that which could be expected from a beginning student.

The outline text editor (EDETS) is now complete and undergoing exhaustive testing. Manuals outlining its use are being written.

The assembler (ASSETS) is so far the only software which is not nearing completion. A somewhat restrictive version of DEC's PAL-11A is being adapted to the system for temporary use. The final assembler for ETS is still being designed. We hope to begin coding in about a month.

The interpretative debugging package (DEBETS) is now operational on a "stand-alone" basis. The necessary linking routines which will load DEBETS and a user program and link them together have been written and are being debugged. The interpreter is a very powerful routine which not only protects the system from the user (the PDP-11 has no relocation or memory protection hardware) but also provides the user with a complete interactive debugging package which provides break points, a full trace mode, a disassembler, symbolic referencing, dump facilities and monitoring facilities for variables or tables.

(D. Oxley)

12.3 Computer Aided Instruction

A CAI program (GIZMO) which was written by CS 201 students in the spring of 1971 is fully operational. However, work is proceeding on a new version, again as a project for CS 201 students.

(A. Davis)

12.4 Publications

Three papers have been submitted and accepted for the ACM SIGPLAN Conference on Pedagogic Languages and to be held in January 1972. These are: "The Use of Mini-Computers for Teaching Assembly Language and System Programming," D. B. Gillies, J. D. Miller, Proceedings of the SIGPLAN Symposium on January 1972 (to appear); "Instructional Software for an Assembly Language Course Taught on a Mini-Computer," D. B. Gillies, et. al., Proceedings of the SIGPLAN Symposium on January 1972 (to appear); and "ETS - An Educational Timesharing System for the PDP-11," D. B. Gillies, T. T. Chen, D. W. Oxley, Proceedings of the SIGPLAN Symposium on January 1972 (to appear).

(A. Davis, D. Gillies, J. Hart
S. Kerr, D. Oxley, D. Pasta, K. Phillips)

12.5 PDP-11 Hardware Additions

The Data Products 2410 line printer which was donated to the project by DEC arrived in late August. It was promptly interfaced and became operational within four days of delivery. A 200 cpm GDI card reader (also donated by DEC) arrived in late September complete with all necessary interface boards. It has performed well even under a production environment requiring continuous operation. The addition of these two peripherals and better operating systems have significantly improved throuput and quality of service.

The 4K Ampex memory interface was completed and is working well. Total core is now 12K words. Design and construction of a ROM module for the PDP-11 and a ROM programmer is underway.

The teletype situation has improved but is still less than ideal. A persistent transient fault was eliminated with a modification to the TTY interrupt PC boards. A test jig is being built for off line testing of the PC boards. Nine TTY's are now on line and being used in testing of ETS Before his retirement, Mr. Huffman built a prototype sound insulating TTY case. Twelve duplicates were built and delivered by the carpenter shop. The cases will be ready for use as soon as cooling fans can be installed.

(J. Miller)

12.6 General Programming

During this period, extensive experimentation with DEC's Disk Operating System (DOS) for the PDP-11 led to modifications which allowed programs to dynamically invoke other programs. Various other modules were replaced in DOS to allow access to local peripherals. A routine to speed up system loading for DOS and other systems was developed (including DECtape copying facilities).

Hardware diagnostics developed include a multi-station teletype noise generator and a line printer tester.

Debugging aids written include a trace routine and an anti-assembler routine (both are operating system independent).

(R. Atkinson)

12.7 SPOOLING

SPOOLING, a batch system for the PDP-11, became operational during this quarter. This system provides fast turnaround through spooling of all card decks and listed output to disk, simple I/O calls for reading data cards and printing lines, user alterable job time and line limits, and debugging aids in the form of core and register dumps. Using the new peripherals (a 200 cpm card reader and 500 lpm line printer), some 750 student jobs have been run in a 15 hour period without saturation and without assistance from the 360/75.

(J. Miller)

12.8 Assembly Language Instruction

Considerable work has been done to improve the structure and resources of CS 201, the introductory course in assembly language and system programming. Major goals and course content have been established, and a book of instructional problems for the course has been written.

The major goal of the course is to teach the students how to write good programs. By stressing such programming principles as planning, documentation, debugging and optimization, as opposed to simply "working code," we help the student develop a knowledge of what good code is, as well as useful techniques.

The course is divided into three sections: basic programming skills, advanced programming techniques, and a study of assemblers. Basic programming includes PAL order code, addressing modes, simple data structures, stacks, and internal documentation. Advanced programming techniques cover program structure, subroutine, debugging, internal and external documentation, optimization, tricks, I/O, traps and interrupts. Both top down and bottom up approaches to assemblers are taught in the last third of the course.

The CS 201 Problem Book is a collection of problems suitable for homework, quizzes, and in class discussion. It covers simple problems useful in the first third of the course, as well as more complex problems and suggested machine problems. The book has been bound and distributed to all students and staff. Answers to approximately half of the problems are given in the back of the book. A complete solutions book has been assembled for the teaching staff. As we collect more problems, the Problem Book will be expanded and revised.

(M. Stone, A. Davis, S. Hansen)

13. GENERAL DEPARTMENT INFORMATION

13.1 Personnel

The number of people associated with the Department in various capacities is given in the following table:

	Full-time	Part-time	Full-time Equivalent
Faculty	24	4	26.27
Visiting Faculty	2	0	2.00
Graduate Research Assistants	1	68	34.50
Graduate Teaching Assistants	0	27	13.25
Professional Personnel	10	1	10.50
Administrative and Clerical	20	0	20.00
Nonacademic Personnel (Monthly)	19	1	19.50
Nonacademic Personnel (Hourly)	0	50	17.60
TOTAL	76	151	143.62

*This report does not include personnel employed on the ILLIAC IV Project.

The Department Advisory Committee consists of Professor J. N. Snyder, Head of the Department, Professors E. K. Bowdon, D. F. Cudia, K. W. Dickman, M. F. Faiman, H. G. Friedman, C. W. Gear, D. B. Gillies, W. J. Kubitz, D. J. Kuck, B. H. McCormick, R. G. Montanelli, S. Muroga, T. A. Murrell, J. Nievergelt, J. R. Phillips, W. J. Poppelbaum, S. R. Ray, E. M. Reingold, J. E. Robertson, P. E. Saylor, D. L. Slotnick, J. E. Vander Mey, D. S. Watanabe, and T. Wilcox.

13.2 Bibliography

During the third quarter, the following publications were issued by the Laboratory:

File Numbers

No. 863 Borovec,R. T., "IMAGE 8: The Real Time Clock", June 16, 1971.

No. 864 McCormick, B. H., "ILLIAC III Bibliography Condensed Listing", July 13, 1971

No. 865 Haskin, R., "GRAPHICS 8: SYSTEM MAINTENANCE (SOFTWARE)", July 1971.

No. 866 Nickolls, J. and Michel, M. J., "GRAPHICS 8: System Maintenance (Hardware)", August 1971.

No. 867 Michel, M. J. and Haskin, R.,"GRASS: Extended Remote Facilities Guide", August 1971.

No. 868 Lewis, G. T., "IMAGE 8: Elliott Paper Tape Reader Control", August 10, 1971.

Report Numbers

No. 250 Knapp, M. and McIntyre, D., "Bulk Storage Applications in the ILLIAC IV System", August 3, 1971.

No. 252 Abel, N., "ICL Your Fancy", September 1, 1971.

No. 465 Michel, M. J., "GRASS: SYSTEM OVERVIEW", July 1971.

No. 466 Haskin, R., Nickolls, J., and Michel, M., "GRASS: Remote Facilities Guide", July 1971.

No. 467 Michel, M. J., and Kock, J., "GRASS: Terminal User's Guide", August 1971.

No. 468 Michel, M. J., "GRASS: System Software Description", August 197

No. 469 Schwebel, J. C., (Revised by Ahmad E. Masumi), "IBAL MANUAL: The Illiac III Basic Assembler Language", July 1971.

No. 470 Nordmann, B. J., Jr., "SPEECH DISPLAY SIMULATION SYSTEM FOR A COMPARATIVE STUDY OF SOME VISUAL SPEECH DISPLAYS", August 197

No. 471 Nakagawa, T., Lai, H.C., and Muroga, S., "PRUNING AND BRANCHING METHODS FOR DESIGNING OPTIMAL NETWORKS BY THE BRANCH-AND-BOUND METHOD", August 1971.

Bibliography (cont.)

No. 472 McCormick, B. H., Nordmann, B. J., Jr., Atkins, D. E., Borovec, R. T., Goyal, L. N., Katoh, L. M., Lansford, R. M., Schwebel, J. C., and Tareski, V. G., "ILLIAC III REFERENCE MANUAL VOLUME IV: Supervisor Organization", August 12, 1971.

No. 473 McCormick, B. H., Nordmann, B. J., Jr., Atkins, D. E., Borovec, R. T., Goyal, L. N., Katoh, L. M., Lansford, R. M., Schwebel, J. C., and Tareski, V. G., "ILLIAC III REFERENCE MANUAL VOLUME IV: Supervisor Organization", August 12, 1971.

No. 475 Nordmann, B. J., Jr., "ILLIAC III COMPUTER SYSTEM MANUAL: TAXICRINIC PROCESSOR VOLUME 2", August 24, 1971.

No. 476 Nordmann, B. J., Jr., "ILLIAC III COMPUTER SYSTEMS MANUAL: TAXICRINIC PROCESSOR VOLUME 3", August 24, 1971.

No. 477 Diamond, M., "A FACTORIZATION PROCEDURE THAT IS EQUIVALENT TO SUCCESSIVE OVER RELAXATION", September 1971.

No. 478 Diamond, M., "THE DIVERGENCE OF STONE'S FACTORIZATIONS WHEN NO PARAMETERS ARE USED", September 1971.

Theses

No. --- Kelly, M. W., "Reliability of Computer Systems ", (M.S.), August 1971. (To be published).

No. 464 Partridge, R.L., "PAGAN: A THREE DIMENSIONAL PATTERN GENERATOR", August 1971.

No. 474 Alster, J. M., "HEURISTIC ALGORITHMS FOR CONSTRUCTING NEAR-OPTIMAL DECISION TREES", August 17, 1971, (M.S.).

No. 479 Nordmann, Jr., B.J., "A COMPARATIVE STUDY OF SOME VISUAL SPEECH DISPLAYS", September 10, 1971, (PhD.).

No. 480 Culliney, J. N., "ON THE SYNTHESIS BY INTEGER PROGRAMMING OF OPTIMAL NOR GATE NETWORKS FOR FOUR VARIABLE SWITCHING FUNCTIONS", September 1971, (M.S.).

No. 485 Kodimer, D. A., "ALPHECON: EVALUATION OF A DEVELOPMENTAL VIDEO STORAGE TUBE", September 1971, (M.S.).

13.3 Colloquia

" The Distributed Computer System", by Professor David J. Farber, Department of Information and Computer Science, University of California at Irvine, Irvine, California 92664, September 27, 1971.

13.4 Drafting

During the third quarter, a total of 234 drawings were processed by the general departmental drafting section:

Large Drawings	96
Medium Drawings	27
Small Drawings	68
Layouts	0
Report Drawings	20
Change Order Drawings	14
Miscellaneous	9
Total	234

(M. Goebel)

13.5 Shop's Production

Job orders processed and completed during the third quarter of 1971 are as follows:

	AEC 2118	AEC 1469	Other
Machine Shop		15	3
Electronic Shop	2	64	7
Chemical Shop	2	65	6
Layout Shop		61	5

(F. P. Serio)

DEPARTMENT OF COMPUTER SCIENCE
GRADUATE COLLEGE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

REPORT REQUEST FORM

Report Number	Title
______	____
______	____
______	____
______	____
______	____

Fold and Staple as shown

NAME: ____

ADDRESS: ____

____

____

Fill out Mailing Label

CUT ALONG THIS LINE

NAME: ____

ADDRESS: ____

____

____

FOLD

Stamp

Mail Room
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

FOLD

STAPLE